Claims

1. A mobile device (62) comprising:

    an application platform having application programming interfaces (APIs), each having a signature identifier;
    a verification system for authenticating digital signatures (96) and signature identifications (94) provided by respective software applications (66) to access the APIs; and
    a control system for allowing a software application on the device to access at least one of the APIs where the signature identifier of the such API corresponds with the digital signature identification and digital signature provided by the software application is authenticated by the verification system.

2. The mobile device of claim 1, wherein a virtual machine comprises the verification system and the control system, preferably the virtual machine being a Java virtual machine and the software application being a Java application.

3. The mobile device claim 1 or 2, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

4. The mobile device of any of claims 1 to 3, wherein the APIs of the application platform access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

5. The mobile device of any of claims 1 to 4, wherein the digital signature is generated using a private signature key under the signature scheme, and the verification system uses a public signature key to authenticate the digital signature.

6. The mobile device of claim 5, wherein:

    the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and
    the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

7. The mobile device of any of claims 1 to 6, wherein at least one of the APIs further comprises:

    a description string that is displayed when the software application attempts to access said at least one of the APIs.

8. A method of controlling access to application programming interfaces (APIs) on a mobile device, including the step of allowing a software application on the device to access at least one of the APIs where a signature identifier of the such API corresponds with a digital signature identification and digital signature provided by the software application is authenticated by a verification system of the device.

9. The method of claim 8, comprising the additional step of:

    purging the software application from the mobile device where the signature identification does not correspond with the signature identifier.

10. The method of claim 8 or 9, wherein the digital signature and the signature identification are generated by a code signing authority.

11. The method of any of claims 8 to 10, comprising the additional step of:

    denying the software application access to the API where the digital signature is not authenticated.

12. The method of claim 11, comprising the additional step of:

    purging the software application from the mobile device where the digital signature is not authenticated.

13. The method of any of claims 8 to 12, wherein a description string is displayed to a user when the software application attempts to access said at least one of the APIs.

14. The method of any of claims 8 to 13 comprising the additional step of:

    displaying a description string that notifies a user of the mobile device that the software application requires access to the API.

15. The method of claim 14, comprising the additional step of:

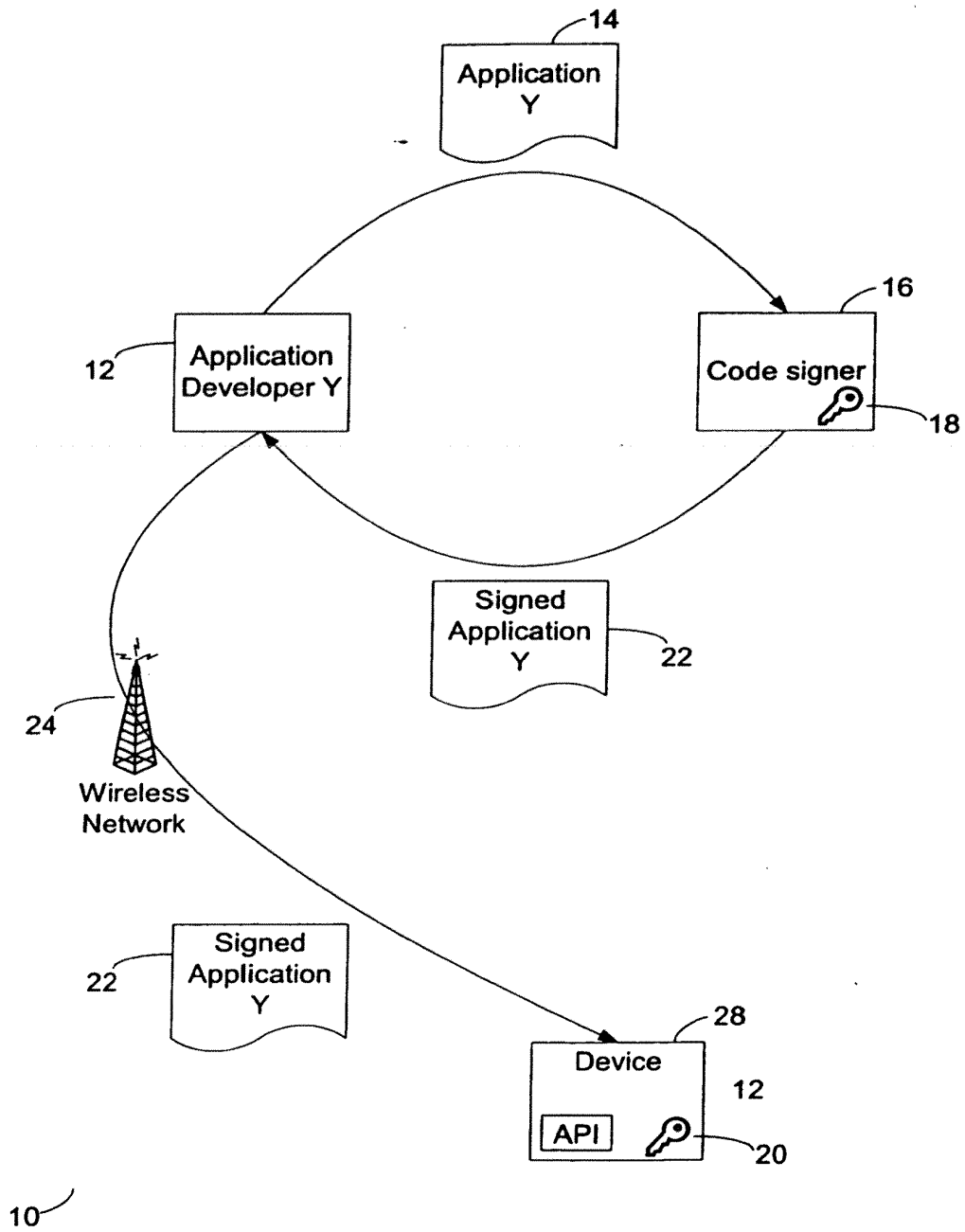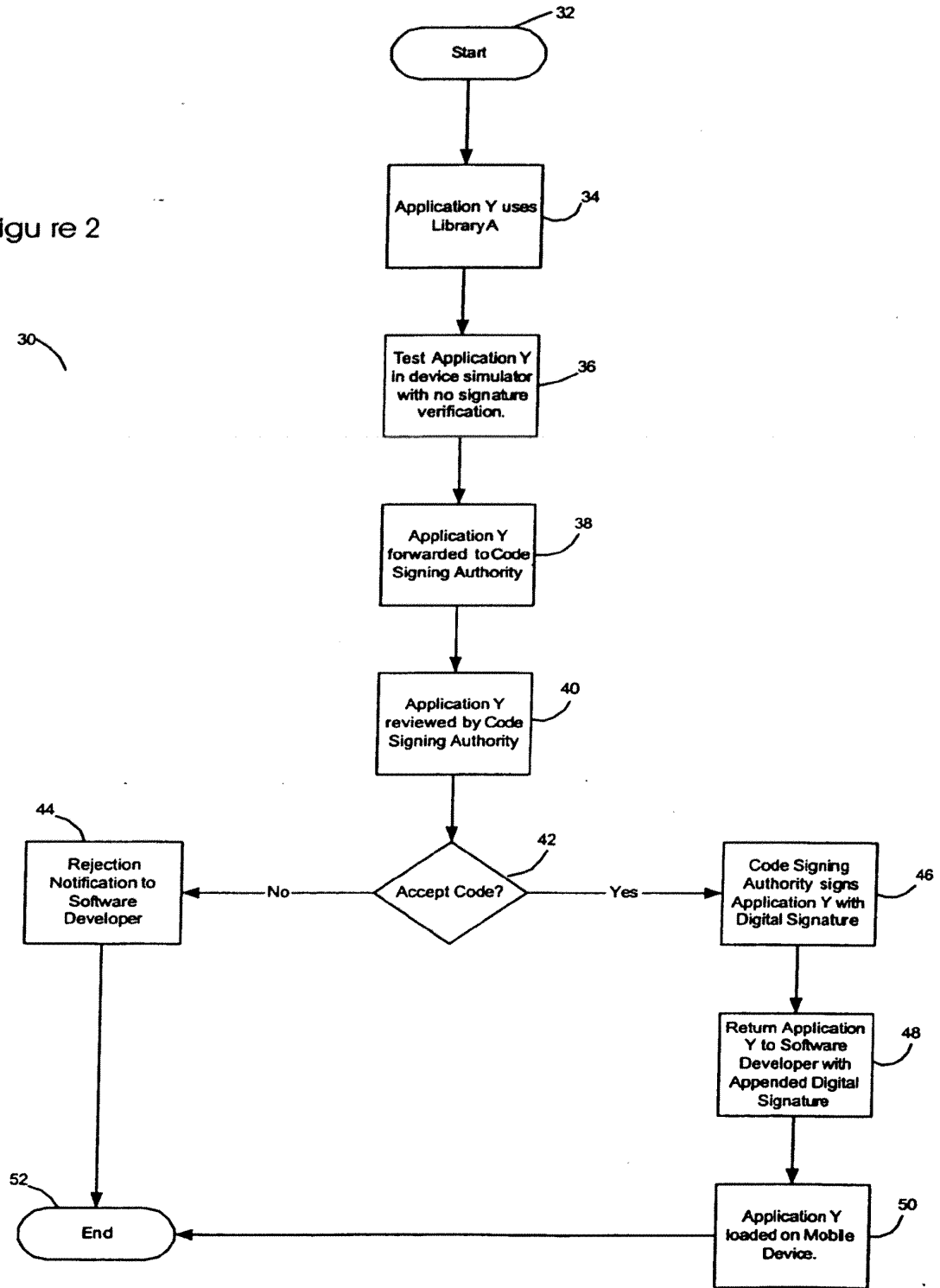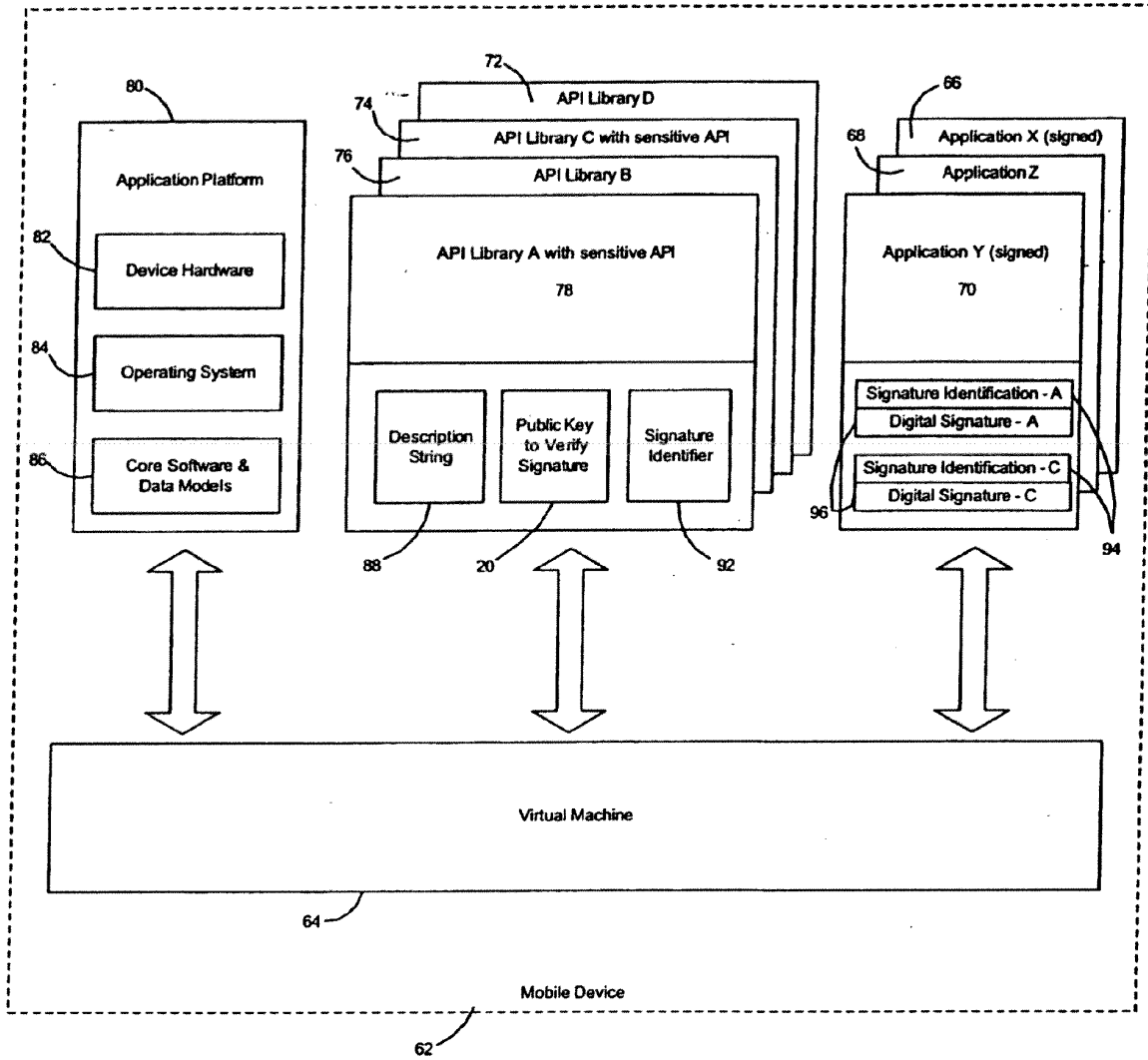   receiving a command from the user granting or denying the software application access to the API.

Figure 1

Figure 2

Figure 3

Figure 3A

102 — Application Loaded
on Mobile Device

104 — Does Application
Need Access to Sensitive
API Library? —No—

Figure 4

100

Yes

106 — Virtual Machine
Retrieves Public
Key and Signature
Identifier from API
Library

108 — Proper
Signature on
Application?

—No—

Yes

110 — Signature
Verified?

—No—

Yes

112 — User Prompted

116 — Application Not
Loaded or
Executed

—No— 114 — Execute
Application?

Yes

118 — Virtual Machine
executes
Application and
linkds with API
Library

120 — End

19

210

Application
Developed

220

Receive Target
Signing Request

235

Developer
Database

230

Developer
Trusted by
Authority?

250

N → Reject Application

Y

245

Target
Private Key
Database

240

Have Target
Key?

270

N → Reject Application

Y

260

Sign Application

280

Return
Signature

200

Figure 5

20

Figure 6

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

EUROPEAN SEARCH REPORT

	DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	ANONYMOUS: "ISO/IEC 7816-4:1995 Information technology -- Identification cards -- Integrated circuit(s) cards with contacts -- Part 4: Interindustry commands for interchange", INTERNATIONAL STANDARD ISO/IEC, vol. 7816-4:1995(E), 1 January 1995 (1995-01-01), pages I-IV,1-46, XP008124701, * page 12 *	1-15	INV. G06F1/00
A	ANONYMOUS: "ISO/IEC 7816-8: IDENTIFICATION CARDS -- INTEGRATED CIRCUIT CARDS - PART 8: COMMANDS FOR SECURITY OPERATIONS", INTERNATIONAL STANDARD ISO/IEC, vol. 7816, no. 8, 25 June 1998 (1998-06-25), XP002610578, * Document consists of pages i-iii, 2-3, 6-13 * * table 4 *	1-15	
A	ANONYMOUS: "ISO/IEC 7816-9: IDENTIFICATION CARDS -- INTEGRATED CIRCUIT CARDS - PART 9: COMMANDS FOR CARD MANAGEMENT", INTERNATIONAL STANDARD ISO/IEC, vol. 7816, no. 9, 17 June 1999 (1999-06-17), XP002610579, * Document consists of i-iv, 9-13, 29-31 * * page 9 * -/--	1-15	TECHNICAL FIELDS SEARCHED (IPC) G06F

The present search report has been drawn up for all claims

3

Place of search	Date of completion of the search	Examiner
Munich	25 November 2010	Kerschbaumer, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another
    document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or
    after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding
    document

EPO FORM 1503 03.82 (P04C01)

22

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

EUROPEAN SEARCH REPORT

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	"Excerpts ED - RANKL W; EFFING W", 1 January 1999 (1999-01-01), HANDBUCH DER CHIPKARTEN. AUFBAU - FUNKTIONSWEISE - EINSATZ VON SMART CARDS, MUENCHEN : CARL HANSER VERLAG, DE, XP007908384, ISBN: 978-3-446-21115-5 * Document concicts of pages 197-203, 261-273, 740-741, 794-797 * * pages 269,272; figure 5.39 * -----	1-15	

TECHNICAL FIELDS
SEARCHED   (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	25 November 2010	Kerschbaumer, J

3

EPO FORM 1503 03.82 (P04C01)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 23415200 P **[0001]**
- US 23535400 P **[0001]**
- US 27066301 P **[0001]**

(54)   **Software code signing system and method**

(57)   A code signing system and method is provided.
The code signing system operates in conjunction with a
signed software application having a digital signature and
includes an application platform, an application program-
ming interface (API), and a virtual machine. The API is
configured to link the software application with the appli-
cation platform. The virtual machine verifies the authen-
ticity of the digital signature in order to control access to
the API by the software application.

Figure 1

EP 2 278 429 A1

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001]   This application claims priority from and is related to the following prior applications: "Code Signing System And Method," U.S. Provisional Application No. 60/234,152, filed Sep. 21, 2000; "Code Signing System And Method," U.S. Provisional Application No. 60/235,354, filed Sep. 26, 2000; and "Code Signing System And Method," U.S. Provisional Application No. 60/270,663, filed Feb. 20,2001.

BACKGROUND

1. Field of the Invention

[0002]   This invention relates generally to the field of security protocols for software applications. More particularly, the invention provides a code signing system and method that is particularly well suited for Java(TM) applications for mobile communication devices, such as Personal Digital Assistants, cellular telephones, and wireless two-way communication devices (collectively referred to hereinafter as "mobile devices" or simply "devices").

2. Description of the Related Art

[0003]   Security protocols involving software code signing schemes are known. Typically, such security protocols are used to ensure the reliability of software applications that are downloaded from the Internet. In a typical software code signing scheme, a digital signature is attached to a software application that identifies the software developer. Once the software is downloaded by a user, the user typically must use his or her judgment to determine whether or not the software application is reliable, based solely on his or her knowledge of the software developer's reputation. This type of code signing scheme does not ensure that a software application written by a third party for a mobile device will properly interact with the device's native applications and other resources. Because typical code signing protocols are not secure and rely solely on the judgment of the user, there is a serious risk that destructive, "Trojan horse" type software applications may be downloaded and installed onto a mobile device.

[0004]   There also remains a need for network operators to have a system and method to maintain control over which software applications are activated on mobile devices.

[0005]   There remains a further need in 2.5G and 3G networks where corporate clients or network operators would like to control the types of software on the devices issued to its employees.

SUMMARY

[0006]   A code signing system and method is provided. The code signing system operates in conjunction with a software application having a digital signature and includes an application platform, an application programming interface (API), and a virtual machine. The API is configured to link the software application with the application platform. The virtual machine verifies the authenticity of the digital signature in order to control access to the API by the software application.
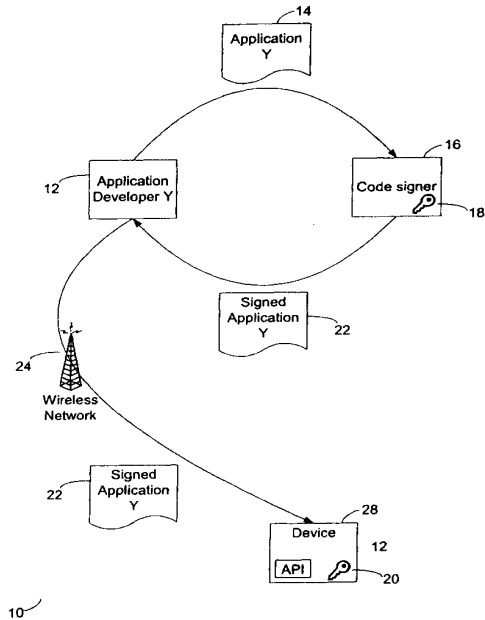
[0007]   A code signing system for operation in conjunction with a software application having a digital signature, according to another embodiment of the invention comprises an application platform, a plurality of APIs, each configured to link the software application with a resource on the application platform, and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0008]   According to a further embodiment of the invention, a method of controlling access to sensitive application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to a sensitive API, determining whether or not the software application includes a digital signature associated with the sensitive API, and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0009]   In another embodiment of the invention, a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer comprises the steps of receiving the software application from the software developer, reviewing the software application to determine if it may access the API, if the software application may access the API, then appending a digital signature to the software application, verifying the authenticity of a digital signature appended to a software application, and providing access to the API to software applications for which the appended digital signature is authentic.

[0010]   A method of restricting access to a sensitive API on a mobile device, according to a further embodiment of the invention, comprises the steps of registering one or more software developers that are trusted to design software applications which access the sensitive API, receiving a hash of a software application, determining if the software application was designed by one of the registered software developers, and if the software application was designed by one of the registered software developers, then generating a digital signature using the hash of the software application, wherein the digital signature may be appended to the software application, and the mobile device verifies the authenticity of the

digital signature in order to control access to the sensitive API by the software application.

[0011] In a still further embodiment, a method of restricting access to application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to one or more API, determining whether or not the software application includes a digital signature associated with the mobile device, and if the software application does not include a digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012]

FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention;
FIG. 2 is a flow diagram of the code signing protocol described above with reference to FIG. 1;
FIG. 3 is a block diagram of a code signing system on a mobile device;
FIG. 3A is a block diagram of a code signing system on a plurality of mobile devices;
FIG. 4 is a flow diagram illustrating the operation of the code signing system described above with reference to FIG. 3 and FIG. 3A;
FIG. 5 is a flow diagram illustrating the management of the code signing authorities described with reference to FIG. 3A; and
FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented.

DETAILED DESCRIPTION

[0013] Referring now to the drawing figures, FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention. An application developer 12 creates a software application 14 (application Y) for a mobile device that requires access to one or more sensitive APIs on the mobile device. The software application Y 14 may, for example, be a Java application that operates on a Java virtual machine installed on the mobile device. An API enables the software application Y to interface with an application platform that may include, for example, resources such as the device hardware, operating system and core software and data models. In order to make function calls to or otherwise interact with such device resources, a software application Y must access one or more APIs. APIs can thereby effectively "bridge" a software application and associated device resources. In this description and the appended claims, references to API access should be interpreted to include access of an API in such a way as to allow a software application Y to interact with one or more corresponding device resources. Providing access to any API therefore allows a software application Y to interact with associated device resources, whereas denying access to an API prevents the software application Y from interacting with the associated resources. For example, a database API may communicate with a device file or data storage system, and access to the database API would provide for interaction between a software application Y and the file or data storage system. A user interface (UI) API would communicate with controllers and/or control software for such device components as a screen, a keyboard, and any other device components that provide output to a user or accept input from a user. In a mobile device, a radio API may also be provided as an interface to wireless communication resources such as a transmitter and receiver. Similarly, a cryptographic API may be provided to interact with a crypto module which implements crypto algorithms on a device. These are merely illustrative examples of APIs that may be provided on a device. A device may include any of these example APIs, or different APIs instead of or in addition to those described above.

[0014] Preferably, any API may be classified as sensitive by a mobile device manufacturer, or possibly by an API author, a wireless network operator, a device owner or operator, or some other entity that may be affected by a virus or malicious code in a device software application. For instance, a mobile device manufacturer may classify as sensitive those APIs that interface with cryptographic routines, wireless communication functions, or proprietary data models such as address book or calendar entries. To protect against unauthorized access to these sensitive APIs, the application developer 12 is required to obtain one or more digital signatures from the mobile device manufacturer or other entity that classified any APIs as sensitive, or from a code signing authority 16 acting on behalf of the manufacturer or other entity with an interest in protecting access to sensitive device APIs, and append the signature(s) to the software application Y 14.

[0015] In one embodiment, a digital signature is obtained for each sensitive API or library that includes a sensitive API to which the software application requires access. In some cases, multiple signatures are desirable. This would allow a service provider, company or network operator to restrict some or all software applications loaded or updated onto a particular set of mobile devices. In this multiple-signature scenario, all APIs are restricted and locked until a "global" signature is verified for a software application. For example, a company may wish to prevent its employees from executing any software applications onto their devices without first obtaining permission from a corporate information technology (IT) or computer services department. All such corporate mobile devices may then be configured to require verification of at least a global signature before a software application can be executed. Access to sensitive device APIs and libraries, if any, could then be further restricted, dependent upon verification of respective corresponding digital signatures.

[0016] The binary executable representation of software application Y 14 may be independent of the particular type of mobile device or model of a mobile device. Software application Y 14 may for example be in a write-once-run-anywhere binary format such as is the case with Java software applications. However, it may be desirable to have a digital signature for each mobile device type or model, or alternatively for each mobile device platform or manufacturer. Therefore, software application Y 14 may be submitted to several code signing authorities if software application Y 14 targets several mobile devices.

[0017] Software application Y 14 is sent from the application developer 12 to the code signing authority 16. In the embodiment shown in FIG. 1, the code signing authority 16 reviews the software application Y 14, although as described in further detail below, it is contemplated that the code signing authority 16 may also or instead consider the identity of the application developer 12 to determine whether or not the software application Y 14 should be signed. The code signing authority 16 is preferably one or more representatives from the mobile device manufacturer, the authors of any sensitive APIs, or possibly others that have knowledge of the operation of the sensitive APIs to which the software application needs access.

[0018] If the code signing authority 16 determines that software application Y 14 may access the sensitive API and therefore should be signed, then a signature (not shown) for the software application Y 14 is generated by the code signing authority 16 and appended to the software application Y 14. The signed software application Y 22, comprising the software application Y 14 and the digital signature, is then returned to the application developer 12. The digital signature is preferably a tag that is generated using a private signature key 18 maintained solely by the code signing authority 16. For example, according to one signature scheme, a hash of the software application Y 14 may be generated, using a hashing algorithm such as the Secure Hash Algorithm SHA1, and then used with the private signature key 18 to create the digital signature. In some signature schemes, the private signature key is used to encrypt a hash of information to be signed, such as software application Y 14, whereas in other schemes, the private key may be used in other ways to generate a signature from the information to be signed or a transformed version of the information.

[0019] The signed software application Y 22 may then be sent to a mobile device 28 or downloaded by the mobile device 28 over a wireless network 24. It should be understood, however, that a code signing protocol according to the present invention is not limited to software applications that are downloaded over a wireless network. For instance, in alternative embodiments, the signed software application Y 22 may be downloaded to a personal computer via a computer network and loaded to the mobile device through a serial link, or may be acquired from the application developer 12 in any other

manner and loaded onto the mobile device. Once the signed software application Y 22 is loaded on the mobile device 28, each digital signature is preferably verified with a public signature key 20 before the software application Y 14 is granted access to a sensitive API library. Although the signed software application Y 22 is loaded onto a device, it should be appreciated that the software application that may eventually be executed on the device is the software application Y 14. As described above, the signed software application Y 22 includes the software application Y 14 and one or more appended digital signatures (not shown). When the signatures are verified, the software application Y 14 can be executed on the device and access any APIs for which corresponding signatures have been verified.

[0020] The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority 16, and is preferably installed on the mobile device along with the sensitive API. However, the public key 10 may instead be obtained from a public key repository (not shown), using the device 28 or possibly a personal computer system, and installed on the device 28 as needed. According to one embodiment of a signature scheme, the mobile device 28 calculates a hash of the software application Y 14 in the signed software application Y 22, using the same hashing algorithm as the code signing authority 16, and uses the digital signature and the public signature key 20 to recover the hash calculated by the signing authority 16. The resultant locally calculated hash and the hash recovered from the digital signature are then compared, and if the hashes are the same, the signature is verified. The software application Y 14 can then be executed on the device 28 and access any sensitive APIs for which the corresponding signature(s) have been verified. As described above, the invention is in no way limited to this particular illustrative example signature scheme. Other signature schemes, including further public key signature schemes, may also be used in conjunction with the code signing methods and systems described herein.

[0021] FIG. 2 is a flow diagram 30 of the code signing protocol described above with reference to FIG. 1. The protocol begins at step 32. At step 34, a software developer writes the software application Y for a mobile device that requires access to a sensitive API or library that exposes a sensitive API (API library A). As discussed above, some or all APIs on a mobile device may be classified as sensitive, thus requiring verification of a digital signature for access by any software application such as software application Y. In step 36, application Y is tested by the software developer, preferably using a device simulator in which the digital signature verification function has been disabled. In this manner, the software developer may debug the software application Y before the digital signature is acquired from the code signing authority. Once the software application Y has been written and debugged, it is forwarded to the code signing authority in step 38.

Page 515 of 1415

[0022] In steps 40 and 42, the code signing authority reviews the software application Y to determine whether or not it should be given access to the sensitive API, and either accepts or rejects the software application. The code signing authority may apply a number of criteria to determine whether or not to grant the software application access to the sensitive API including, for example, the size of the software application, the device resources accessed by the API, the perceived utility of the software application, the interaction with other software applications, the inclusion of a virus or other destructive code, and whether or not the developer has a contractual obligation or other business arrangement with the mobile device manufacturer. Further details of managing code signing authorities and developers are described below in reference to FIG. 5.

[0023] If the code signing authority accepts the software application Y, then a digital signature, and preferably a signature identification, are appended to the software application Y in step 46. As described above, the digital signature may be generated by using a hash of the software application Y and a private signature key 18. The signature identification is described below with reference to FIGS. 3 and 4. Once the digital signature and signature identification are appended to the software application Y to generate a signed software application, the signed software application Y is returned to the software developer in step 48. The software developer may then license the signed software application Y to be loaded onto a mobile device (step 50). If the code signing authority rejects the software application Y, however, then a rejection notification is preferably sent to the software developer (step 44), and the software application Y will be unable to access any API(s) associated with the signature.

[0024] In an alternative embodiment, the software developer may provide the code signing authority with only a hash of the software application Y, or provide the software application Y in some type of abridged format. If the software application Y is a Java application, then the device independent binary *.class files may be used in the hashing operation, although device dependent files such as *.cod files used by the assignee of the present application may instead be used in hashing or other digital signature operations when software applications are intended for operation on particular devices or device types. By providing only a hash or abridged version of the software application Y, the software developer may have the software application Y signed without revealing proprietary code to the code signing authority. The hash of the software application Y, along with the private signature key 18, may then be used by the code signing authority to generate the digital signature. If an otherwise abridged version of the software application Y is sent to the code signing authority, then the abridged version may similarly be used to generate the digital signature, provided that the abridging scheme or algorithm, like a hashing algorithm, generates different outputs for different inputs. This ensures that every software application will have a different abridged version and thus a different signature that can only be verified when appended to the particular corresponding software application from which the abridged version was generated. Because this embodiment does not enable the code signing authority to thoroughly review the software application for viruses or other destructive code, however, a registration process between the software developer and the code signing authority may also be required. For instance, the code signing authority may agree in advance to provide a trusted software developer access to a limited set of sensitive APIs.

[0025] In still another alternative embodiment, a software application Y may be submitted to more than one signing authority. Each signing authority may for example be responsible for signing software applications for particular sensitive APIs or APIs on a particular model of mobile device or set of mobile devices that supports the sensitive APIs required by a software application. A manufacturer, mobile communication network operator, service provider, or corporate client for example may thereby have signing authority over the use of sensitive APIs for their particular mobile device model(s), or the mobile devices operating on a particular network, subscribing to one or more particular services, or distributed to corporate employees. A signed software application may then include a software application and at least one appended digital signature appended from each of the signing authorities. Even though these signing authorities in this example would be generating a signature for the same software application, different signing and signature verification schemes may be associated with the different signing authorities.

[0026] FIG. 3 is a block diagram of a code signing system 60 on a mobile device 62. The system 60 includes a virtual machine 64, a plurality of software applications 66-70, a plurality of API libraries 72-78, and an application platform 80. The application platform 80 preferably includes all of the resources on the mobile device 62 that may be accessed by the software applications 66-70. For instance, the application platform may include device hardware 82, the mobile device's operating system 84, or core software and data models 86. Each API library 72-78 preferably includes a plurality of APIs that interface with a resource available in the application platform. For instance, one API library might include all of the APIs that interface with a calendar program and calendar entry data models. Another API library might include all of the APIs that interface with the transmission circuitry and functions of the mobile device 62. Yet another API library might include all of the APIs capable of interfacing with lower-level services performed by the mobile device's operating system 84. In addition, the plurality of API libraries 72-78 may include both libraries that expose a sensitive API 74 and 78, such as an interface to a cryptographic function, and libraries 72 and 76, that may be accessed without exposing sensitive APIs. Similarly, the

plurality of software applications 66-70 may include both signed software applications 66 and 70 that require access to one or more sensitive APIs, and unsigned software applications such as 68. The virtual machine 64 is preferably an object oriented run-time environment such as Sun Micro System's J2ME(TM) (Java 2 Platform, Micro Edition), which manages the execution of all of the software applications 66-70 operating on the mobile device 62, and links the software applications 66-70 to the various API libraries 72-78.

[0027] Software application Y 70 is an example of a signed software application. Each signed software application preferably includes an actual software application such as software application Y comprising for example software code that can be executed on the application platform 80, one or more signature identifications 94 and one or more corresponding digital signatures 96. Preferably each digital signature 96 and associated signature identification 94 in a signed software application 66 or 70 corresponds to a sensitive API library 74 or 78 to which the software application X or software application Y requires access. The sensitive API library 74 or 78 may include one or more sensitive APIs. In an alternative embodiment, the signed software applications 66 and 70 may include a digital signature 96 for each sensitive API within an API library 74 or 78. The signature identifications 94 may be unique integers or some other means of relating a digital signature 96 to a specific API library 74 or 78, API, application platform 80, or model of mobile device 62.

[0028] API library A 78 is an example of an API library that exposes a sensitive API. Each API library 74 and 78 including a sensitive API should preferably include a description string 88, a public signature key 20, and a signature identifier 92. The signature identifier 92 preferably corresponds to a signature identification 94 in a signed software application 66 or 70, and enables the virtual machine 64 to quickly match a digital signature 96 with an API library 74 or 78. The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority, and is used to verify the authenticity of a digital signature 96. The description string 88 may for example be a textual message that is displayed on the mobile device when a signed software application 66 or 70 is loaded, or alternatively when a software application X or Y attempts to access a sensitive API.

[0029] Operationally, when a signed software application 68-70, respectively including a software application X, Z, or Y, that requires access to a sensitive API library 74 or 78 is loaded onto a mobile device, the virtual machine 64 searches the signed for an appended digital signature 96 associated with the API library 74 or 78. Preferably, the appropriate digital signature 96 is located by the virtual machine 64 by matching the signature identifier 92 in the API library 74 or 78 with a signature identification 94 on the signed software application. If the signed software application includes the appropriate dig-

ital signature 96, then the virtual machine 64 verifies its authenticity using the public signature key 20. Then, once the appropriate digital signature 96 has been located and verified, the description string 88 is preferably displayed on the mobile device before the software application X or Y is executed and accesses the sensitive API. For instance, the description string 88 may display a message stating that "Application Y is attempting to access API Library A," and thereby provide the mobile device user with the final control to grant or deny access to the sensitive API.

[0030] FIG. 3A is a block diagram of a code signing system 61 on a plurality of mobile devices 62E, 62F and 62G. The system 61 includes a plurality of mobile devices each of which only three are illustrated, mobile devices 62E, 62F and 62G. Also shown is a signed software application 70, including a software application Y to which two digital signatures 96E and 96F with corresponding signature identifications 94E and 94F have been appended. In the example system 61, each pair composed of a digital signature and identification, 94E/96E and 94F/96F, corresponds to a model of mobile device 62, API library 78, or associated platform 80. If signature identifications 94E and 94F correspond to different models of mobile device 62, then when a signed software application 70 which includes a software application Y that requires access to a sensitive API library 78 is loaded onto mobile device 62E, the virtual machine 64 searches the signed software application 70 for a digital signature 96E associated with the API library 78 by matching identifier 94E with signature identifier 92. Similarly, when a signed software application 70 including a software application Y that requires access to a sensitive API library 78 is loaded onto a mobile device 62F, the virtual machine 64 in device 62F searches the signed software application 70 for a digital signature 96F associated with the API library 78. However, when a software application Y in a signed software application 70 that requires access to a sensitive API library 78 is loaded onto a mobile device model for which the application developer has not obtained a digital signature, device 62G in the example of FIG. 3A, the virtual machine 64 in the device 64G does not find a digital signature appended to the software application Y and consequently, access to the API library 78 is denied on device 62G. It should be appreciated from the foregoing description that a software application such as software application Y may have multiple device-specific, library-specific, or API-specific signatures or some combination of such signatures appended thereto. Similarly, different signature verification requirements may be configured for the different devices. For example, device 62E may require verification of both a global signature, as well as additional signatures for any sensitive APIs to which a software application, requires access in order for the software application to be executed, whereas device 62F may require verification of only a global signature and device 62G may require verification of signatures only for its sensitive APIs. It should also be ap-

parent that a communication system may include devices (not shown) on which a software application Y received as part of a signed software application such as 70 may execute without any signature verification. Although a signed software application has one or more signatures appended thereto, the software application Y might possibly be executed on some devices without first having any of its signature(s) verified. Signing of a software application preferably does not interfere with its execution on devices in which digital signature verification is not implemented.

[0031] FIG. 4 is a flow diagram 100 illustrating the operation of the code signing system described above with reference to FIGS. 3 and 3A. In step 102, a software application is loaded onto a mobile device. Once the software application is loaded, the device, preferably using a virtual machine, determines whether or not the software application requires access to any API libraries that expose a sensitive API (step 104). If not, then the software application is linked with all of its required API libraries and executed (step 118). If the software application does require access to a sensitive API, however, then the virtual machine verifies that the software application includes a valid digital signature associated any sensitive APIs to which access is required, in steps 106-116.

[0032] In step 106, the virtual machine retrieves the public signature key 20 and signature identifier 92 from the sensitive API library. The signature identifier 92 is then used by the virtual machine in step 108 to determine whether or not the software application has an appended digital signature 96 with a corresponding signature identification 94. If not, then the software application has not been approved for access to the sensitive API by a code signing authority, and the software application is preferably prevented from being executed in step 116. In alternative embodiments, a software application without a proper digital signature 96 may be purged from the mobile device, or may be denied access to the API library exposing the sensitive API but executed to the extent possible without access to the API library. It is also contemplated that a user may be prompted for an input when signature verification fails, thereby providing for user control of such subsequent operations as purging of the software application from the device.

[0033] If a digital signature 96 corresponding to the sensitive API library is appended to the software application and located by the virtual machine, then the virtual machine uses the public key 20 to verify the authenticity of the digital signature 96 in step 110. This step may be performed, for example, by using the signature verification scheme described above or other alternative signature schemes. If the digital signature 96 is not authentic, then the software application is preferably either not executed, purged, or restricted from accessing the sensitive API as described above with reference to step 116. If the digital signature is authentic, however, then the description string 88 is preferably displayed in step 112, warning the mobile device user that the software application re-

quires access to a sensitive API, and possibly prompting the user for authorization to execute or load the software application (step 114). When more than one signature is to be verified for a software application, then the steps 104-110 are preferably repeated for each signature before the user is prompted in step 112. If the mobile device user in step 114 authorizes the software application, then it may be executed and linked to the sensitive API library in step 118.

[0034] FIG. 5 is a flow diagram 200 illustrating the management of the code signing authorities described with reference to FIG. 3A. At step 210, an application developer has developed a new software application which is intended to be executable one or more target device models or types. The target devices may include sets of devices from different manufacturers, sets of device models or types from the same manufacturer, or generally any sets of devices having particular signature and verification requirements. The term "target device" refers to any such set of devices having a common signature requirement. For example, a set of devices requiring verification of a device-specific global signature for execution of all software applications may comprise a target device, and devices that require both a global signature and further signatures for sensitive APIs may be part of more than one target device set. The software application may be written in a device independent manner by using at least one known API, supported on at least one target device with an API library. Preferably, the developed software application is intended to be executable on several target devices, each of which has its own at least one API library.

[0035] At step 220, a code signing authority for one target device receives a target-signing request from the developer. The target signing request includes the software application or a hash of the software application, a developer identifier, as well as at least one target device identifier which identifies the target device for which a signature is being requested. At step 230, the signing authority consults a developer database 235 or other records to determine whether or not to trust developer 220. This determination can be made according to several criteria discussed above, such as whether or not the developer has a contractual obligation or has entered into some other type of business arrangement with a device manufacturer, network operator, service provider, or device manufacturer. If the developer is trusted, then the method proceeds at step 240. However, if the developer is not trusted, then the software application is rejected (250) and not signed by the signing authority. Assuming the developer was trusted, at step 240 the signing authority determines if it has the target private key corresponding to the submitted target identifier by consulting a private key store such as a target private key database 245. If the target private key is found, then a digital signature for the software application is generated at step 260 and the digital signature or a signed software application including the digital signature appended to the software application is returned to the developer at step

280. However, if the target private key is not found at step 240, then the software application is rejected at step 270 and no digital signature is generated for the software application.

[0036] Advantageously, if target signing authorities follow compatible embodiments of the method outlined in FIG. 5, a network of target signing authorities may be established in order to expediently manage code signing authorities and a developer community code signing process providing signed software applications for multiple targets with low likelihood of destructive code.

[0037] Should any destructive or otherwise problematic code be found in a software application or suspected because of behavior exhibited when a software application is executed on a device, then the registration or privileges of the corresponding application developer with any or all signing authorities may also be suspended or revoked, since the digital signature provides an audit trail through which the developer of a problematic software application may be identified. In such an event, devices may be informed of the revocation by being configured to periodically download signature revocation lists, for example. If software applications for which the corresponding digital signatures have been revoked are running on a device, the device may then halt execution of any such software application and possibly purge the software application from its local storage. If preferred, devices may also be configured to re-execute digital signature verifications, for instance periodically or when a new revocation list is downloaded.

[0038] Although a digital signature generated by a signing authority is dependent upon authentication of the application developer and confirmation that the application developer has been properly registered, the digital signature is preferably generated from a hash or otherwise transformed version of the software application and is therefore application-specific. This contrasts with known code signing schemes, in which API access is granted to any software applications arriving from trusted application developers or authors. In the code signing systems and methods described herein, API access is granted on an application-by-application basis and thus can be more strictly controlled or regulated.

[0039] FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented. The mobile communication device 610 is preferably a two-way communication device having at least voice and data communication capabilities. The device preferably has the capability to communicate with other computer systems on the Internet. Depending on the functionality provided by the device, the device may be referred to as a data messaging device, a two-way pager, a cellular telephone with data messaging capabilities, a wireless Internet appliance or a data communication device (with or without telephony capabilities).

[0040] Where the device 610 is enabled for two-way communications, the device will incorporate a communi-

cation subsystem 611, including a receiver 612, a transmitter 614, and associated components such as one or more, preferably embedded or internal, antenna elements 616 and 618, local oscillators (LOs) 613, and a processing module such as a digital signal processor (DSP) 620. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem 611 will be dependent upon the communication network in which the device is intended to operate. For example, a device 610 destined for a North American market may include a communication subsystem 611 designed to operate within the Mobitex(TM) mobile communication system or DataTAC(TM) mobile communication system, whereas a device 610 intended for use in Europe may incorporate a General Packet Radio Service (GPRS) communication subsystem 611.

[0041] Network access requirements will also vary depending upon the type of network 919. For example, in the Mobitex and DataTAC networks, mobile devices such as 610 are registered on the network using a unique identification number associated with each device. In GPRS networks however, network access is associated with a subscriber or user of a device 610. A GPRS device therefore requires a subscriber identity module (not shown), commonly referred to as a SIM card, in order to operate on a GPRS network. Without a SIM card, a GPRS device will not be fully functional. Local or non-network communication functions (if any) may be operable, but the device 610 will be unable to carry out any functions involving communications over network 619, other than any legally required operations such as "911" emergency calling.

[0042] When required network registration or activation procedures have been completed, a device 610 may send and receive communication signals over the network 619. Signals received by the antenna 616 through a communication network 619 are input to the receiver 612, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in FIG. 6, analog to digital conversion. Analog to digital conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP 620. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by the DSP 620 and input to the transmitter 614 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission over the communication network 619 via the antenna 618.

[0043] The DSP 620 not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in the receiver 612 and transmitter 614 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 620.

[0044] The device 610 preferably includes a microprocessor 638 which controls the overall operation of the device. Communication functions, including at least data

8

and voice communications, are performed through the communication subsystem 611. The microprocessor 638 also interacts with further device subsystems or resources such as the display 622, flash memory 624, random access memory (RAM) 626, auxiliary input/output (I/O) subsystems 628, serial port 630, keyboard 632, speaker 634, microphone 636, a short-range communications subsystem 640 and any other device subsystems generally designated as 642. APIs, including sensitive APIs requiring verification of one or more corresponding digital signatures before access is granted, may be provided on the device 610 to interface between software applications and any of the resources shown in FIG. 6.

[0045] Some of the subsystems shown in FIG. 6 perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard 632 and display 622 for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0046] Operating system software used by the microprocessor 638, and possibly APIs to be accessed by software applications, is preferably stored in a persistent store such as flash memory 624, which may instead be a read only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device software applications, or parts thereof, may be temporarily loaded into a volatile store such as RAM 626. It is contemplated that received and transmitted communication signals may also be stored to RAM 626.

[0047] The microprocessor 638, in addition to its operating system functions, preferably enables execution of software applications on the device. A predetermined set of applications which control basic device operations, including at least data and voice communication applications for example, will normally be installed on the device 610 during manufacture. A preferred application that may be loaded onto the device may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the device user such as, but not limited to e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the device to facilitate storage of PIM data items on the device. Such PIM application would preferably have the ability to send and receive data items, via the wireless network. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network, with the device user's corresponding data items stored or associated with a host computer system thereby creating a mirrored host computer on the mobile device with respect to the data items at least. This would be especially advantageous in the case where the host computer system is the mobile device user's office computer system. Further applications, including signed software applications as described above, may also be loaded onto the device 610 through the network 619, an auxiliary I/O subsystem 62S, serial port 630, short-range communications subsystem 640 or any other suitable subsystem 642. The device microprocessor 638 may then verify any digital signatures, possibly including both "global" device signatures and API-specific signatures, appended to such a software application before the software application can be executed by the microprocessor 638 and/or access any associated sensitive APIs. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the device 610, through a crypto API and a crypto module which implements crypto algorithms on the device (not shown).

[0048] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem 611 and input to the microprocessor 638, which will preferably further process the received signal for output to the display 622, or alternatively to an auxiliary I/O device 628. A user of device 610 may also compose data items such as email messages for example, using the keyboard 632, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display 622 and possibly an auxiliary I/O device 628. Such composed items may then be transmitted over a communication network through the communication subsystem 611.

[0049] For voice communications, overall operation of the device 610 is substantially similar, except that received signals would preferably be output to a speaker 634 and signals for transmission would be generated by a microphone 636. Alternative voice or audio I/O subsystems such as a voice message recording subsystem may also be implemented on the device 610. Although voice or audio signal output is preferably accomplished primarily through the speaker 634, the display 622 may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information for example.

[0050] The serial port 630 in FIG. 6 would normally be implemented in a personal digital assistant (PDA)-type communication device for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port 630 would enable a user to set preferences through an external device or software application and would extend the capabilities of the device by providing for information or software downloads to the device 610 other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the device through a direct and thus reliable and trusted connection to thereby enable secure device communication.

[0051] A short-range communications subsystem 640 is a further optional component which may provide for communication between the device 624 and different systems or devices, which need not necessarily be similar devices. For example, the subsystem 640 may include an infrared device and associated circuits and components or a Bluetooth(TM) communication module to provide for communication with similarly-enabled systems and devices.

[0052] The embodiments described herein are examples of structures, systems or methods having elements corresponding to the elements of the invention recited in the claims. This written description may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements of the invention recited in the claims. The intended scope of the invention thus includes other structures, systems or methods that do not differ from the literal language of the claims, and further includes other structures, systems or methods with insubstantial differences from the literal language of the claims.

[0053] For example, when a software application is rejected at step 250 in the method shown in FIG. 5, the signing authority may request that the developer sign a contract or enter into a business relationship with a device manufacturer or other entity on whose behalf the signing authority acts. Similarly, if a software application is rejected at step 270, authority to sign the software application may be delegated to a different signing authority. The signing of a software application following delegation of signing of the software application to the different authority can proceed substantially as shown in FIG. 5, wherein the target signing authority that received the original request from the trusted developer at step 220 requests that the software application be signed by the different signing authority on behalf of the trusted developer from the target signing authority. Once a trust relationship has been established between code signing authorities, target private code signing keys could be shared between code signing authorities to improve performance of the method at step 240, or a device may be configured to validate digital signatures from either of the trusted signing authorities.

[0054] In addition, although described primarily in the context of software applications, code signing systems and methods according to the present invention may also be applied to other device-related components, including but in no way limited to, commands and associated command arguments, and libraries configured to interface with device resources. Such commands and libraries may be sent to mobile devices by device manufacturers, device owners, network operators, service providers, software application developers and the like. It would be desirable to control the execution of any command that may affect device operation, such as a command to change a device identification code or wireless communication network address for example, by requiring verification of one or more digital signatures before a com-mand can be executed on a device, in accordance with the code signing systems and methods described and claimed herein.

[0055] As has been described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; an application programming interface (API) configured to link the software application with the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0056] The virtual machine may deny the software application access to the API if the digital signature is not authentic. The virtual machine may purge the software application if the digital signature is not authentic. The code signing system may be installed on a mobile device. The digital signature may be generated by a code signing authority.

[0057] The code signing system may further comprise a plurality of API libraries each including a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

[0058] One or more of the plurality of API libraries may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive API libraries by the software application. The software application may include a unique digital signature for each sensitive API library. The software application may include a signature identification for each unique digital signature; each sensitive API library may include a signature identifier; and the virtual machine may compare the signature identification and the signature identifier to match the unique digital signatures with sensitive API libraries.

[0059] The digital signature may be generated using a private signature key, and the virtual machine may use a public signature key to verify the authenticity of the digital signature. The digital signature may be generated by applying the private signature key to a hash of the software application; and the virtual machine may verify the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

[0060] The API may further comprise a description string that is displayed by the mobile device when the software application attempts to access the API. The application platform may comprise an operating system. The application platform may comprise one or more core functions of a mobile device. The application platform may comprise hardware on a mobile device. The hardware may comprise a subscriber identity module (SIM) card. The software application may be a Java application for a mobile device. The API may interface with a cryptographic routine on the application platform. The API may interface with a proprietary data model on the application platform. The virtual machine may be a Java

virtual machine installed on a mobile device.

[0061] As also described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; a plurality of application programming interfaces (APIs), each configured to link the software application with a resource on the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0062] The plurality of APIs may be included in an API library. One or more of the plurality of APIs may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive APIs. For operation in conjunction with a plurality of software applications, one or more of the plurality of software applications may have a digital signature, and the virtual machine may verify the authenticity of the digital signature of each of the one or more of the plurality of software applications in order to control access to the sensitive APIs by each of the plurality of software applications. The resource on the application platform may comprise a wireless communication system. The resource on the application platform may comprise a cryptographic module which implements cryptographic algorithms. The resource on the application platform may comprise a data store. The resource on the application platform may comprise a user interface (UI).

[0063] As has also been described, a method of controlling access to sensitive application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to a sensitive application programming interface (API); determining whether or not the software application includes a digital signature associated with the sensitive API; and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0064] The method my comprise the additional step of: if the software application does not include a digital signature associated with the sensitive API, then purging the software application from the mobile device. The digital signature may be generated by a code signing authority. The method may comprise the additional steps of: if the software application includes a digital signature associated with the sensitive API, then verifying the authenticity of the digital signature; and if the digital signature is not authentic, then denying the software application access to the sensitive API. The method may further comprise the additional step of: if the digital signature is not authentic, then purging the software application from the mobile device. The digital signature may be generated by applying a private signature key to a hash of the software application, and the step of verifying the authenticity of the digital signature may be performed by a meth-

od comprising the steps of: storing a public signature key that corresponds to the private signature key on the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash. The digital signature may be generated by calculating a hash of the software application and applying the private signature key. The method may comprise the additional step of: displaying a description string that notifies a user of the mobile device that the software application requires access to the sensitive API. The method may further comprise the additional step of: receiving a command from the user granting or denying the software application access to the sensitive API.

[0065] Further has been described a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer, comprising the steps of: receiving the software application from the software developer; reviewing the software application to determine if it may access the API; if the software application may access the API, then appending a digital signature to the software application; verifying the authenticity of a digital signature appended to a software application; and providing access to the API to software applications for which the appended digital signature is authentic.

[0066] The step of reviewing the software application may be performed by a code signing authority. The step of appending the digital signature to the software application may be performed by a method comprising the steps of: calculating a hash of the software application; and applying a signature key to the hash of the software application to generate the digital signature. The hash of the software application may be calculated using the Secure Hash Algorithm (SHA1). The step of verifying the authenticity of a digital signature may comprise the steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; and determining if the digital signature is authentic by comparing the calculated hash with the recovered hash. The method may further comprise the step of, if the digital signature is not authentic, then denying the software application access to the API. The signature key may be a private signature key and the corresponding signature key is a public signature key.

[0067] Also has been described, a method of controlling access to a sensitive application programming interface (API) on a mobile device, comprising the steps of: registering one or more software developers that are trusted to design software applications which access the sensitive API; receiving a hash of a software application; determining if the software application was designed by one of the registered software developers; and if the software application was designed by one of the registered software developers, then generating a digital signature

using the hash of the software application, wherein the digital signature may be appended to the software application; and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0068] The step of generating the digital signature may be performed by a code signing authority. The step of generating the digital signature may be performed by applying a signature key to the hash of the software application. The mobile device may verify the authenticity of the digital signature by performing the additional steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; determining if the digital signature is authentic by comparing the calculated hash with the recovered hash; and if the digital signature is not authentic, then denying the software application access to the sensitive API.

[0069] As has been described, a method of restricting access to application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to one or more application programming interface (API); determining whether or not the software application includes an authentic digital signature associated with the mobile device; and if the software application does not include an authentic digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

[0070] The method may comprise the additional step of: if the software application does not include an authentic digital signature associated with the mobile device, then purging the software application from the mobile device. The software application may include a plurality of digital signatures. The plurality of digital signatures may include digital signatures respectively associated with different types of mobile devices.

[0071] Each of the plurality of digital signatures may be generated by a respective corresponding code signing authority. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the additional steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature. The one or more APIs may include one or more APIs classified as sensitive, and the method may further comprise the steps of, for each sensitive API: determining whether or not the software application includes an authentic digital signature associated with the sensitive API; and if the software application does not include an authentic digital signature associated with the sensitive API, then denying the software application access to the sensitive API. Each of the plurality of digital signatures may be generated by its corresponding code signing authority by applying a respective private signa-

ture key associated with the code signing authority to a hash of the software application. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature, wherein the step of verifying the authenticity of the digital signature is performed by a method comprising the steps of: storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the signature associated with the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash.

[0072] The following are particularly preferred embodiments according to the present invention.

[0073] Numbered Embodiment 14. A method of controlling access to sensitive application programming interfaces on a mobile device (62), comprising the steps of:

- loading a software application (66) on the mobile device (62) that requires access to a sensitive application programming interface (API) having a signature identifier (92);
- determining whether the software application (66) includes a digital signature (96) and a signature identification (94); and
- denying the software application (66) access to the sensitive API where the signature identification (94) does not correspond with the signature identifier (92).

[0074] Numbered Embodiment 15. The method of Numbered Embodiment 14, comprising the additional step of:

- purging the software application (66) from the mobile device (62) where the signature identification (94) does not correspond with the signature identifier (92).

[0075] Numbered Embodiment 16. The method of Numbered Embodiment 14 or Numbered Embodiment 15, wherein the digital signature (96) and the signature identification (94) are generated by a code signing authority.

[0076] Numbered Embodiment 17. The method of any of Numbered Embodiments 14 to 16, comprising the additional steps of:

- verifying the authenticity of the digital signature (96) where the signature identification (94) corresponds with the signature identifier (92); and
- denying the software application (66) access to the

12

sensitive API where the digital signature (96) is not authenticated.

[0077] Numbered Embodiment 18. The method of Numbered Embodiment 17, comprising the additional step of:

- purging the software application (66) from the mobile device (62) where the digital signature (96) is not authenticated.

[0078] Numbered Embodiment 19. The method of Numbered Embodiment 17, wherein the digital signature (96) is generated by applying a private signature key to a hash of the software application (66), and wherein the step of verifying the authenticity of the digital signature (96) is performed by a method comprising the steps of:

- storing a public signature key that corresponds to the private signature key on the mobile device (62);
- generating a hash of the software application (66) to obtain a generated hash;
- applying the public signature key to the digital signature (96) to obtain a recovered hash; and
- comparing the generated hash with the recovered hash.

[0079] Numbered Embodiment 20. The method of Numbered Embodiment 19, wherein the digital signature (96) is generated by calculating a hash of the software application (66) and applying the private signature key.

[0080] Numbered Embodiment 21. The method of any of Numbered Embodiments 14 to 20, comprising the additional steps of:

- displaying a description string (88) that notifies a user of the mobile device (62) that the software application (66) requires access to the sensitive API.

[0081] Numbered Embodiment 22. The method of Numbered Embodiment 21, comprising the additional step of:

- receiving a command from the user granting or denying the software application (66) access to the sensitive API.

[0082] Numbered Embodiment 23. A mobile device for a mobile device comprising:

- an application platform having application programming interfaces (APIs);
- a verification system for authenticating digital signatures (96) and signature identifications (94) provided by the respective software applications (66) to access the APIs; and
- a control system for allowing a software application (66) to access at least one of the APIs where a digital

signature (96) provided by the software application (66) is authenticated by the verification system;
- wherein a code signing authority provides digital signatures (96) and signature identifications (94) to software applications (66) that require access to at least one of the APIs such that the digital signature (96) for the software application (66) is generated according to a signature scheme of a signature identification (94), and wherein the signature identifications (94) provided to the software applications (66) comprise those signature identifications (94) that are substantially only authorized to allow access on the subset of the plurality of mobile devices (62).

[0083] Numbered Embodiment 24. The mobile device of Numbered Embodiment 23, wherein a virtual machine (64) comprises the verification system and the control system, preferably the virtual machine (64) being a Java virtual machine and the software application being a Java application.

[0084] Numbered Embodiment 25. The mobile device of Numbered Embodiments 23 or 24, wherein the control system requires one digital signature (96) and one signature identification (94) for each library of at least one of the APIs.

[0085] Numbered Embodiment 26. The mobile device of Numbered Embodiments 23 to 25, wherein the APIs of the application platform access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

[0086] Numbered Embodiment 27. The mobile device of Numbered Embodiments 23 to 26, wherein the digital signature (96) is generated using a private signature key under the signature scheme, and the verification system uses a public signature key to authenticate the digital signature.

[0087] Numbered Embodiment 28. The mobile device of Numbered Embodiment 27, wherein:

- the digital signature (96) is generated by applying the private signature key to a hash of the software application (66) under the signature scheme; and
- the verification system authenticates the digital signature (96) by generating a hash of the software application (6) to obtain a generated hash, applying the public signature key to the digital signature (96) to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

[0088] Numbered Embodiment 29. The mobile device of Numbered Embodiments 23 to 28, wherein at least one of the APIs further comprises:

- a description string (88) that is displayed to a user when the software application (66) attempts to access said at least one of the APIs.

Claims

1. A mobile device (62) comprising:

    an application platform having application pro-
    gramming interfaces (APIs), each having a sig-
    nature identifier;
    a verification system for authenticating digital
    signatures (96) and signature identifications
    (94) provided by respective software applica-
    tions (66) to access the APIs; and
    a control system for denying a software applica-
    tion on the device access to at least one of the
    APIs where the signature identifier of the such
    API does not correspond with the digital signa-
    ture identification and digital signature provided
    by the software application or is not authenticat-
    ed by the verification system.

2. The mobile device of claim 1, wherein a virtual ma-
    chine comprises the verification system and the con-
    trol system, preferably the virtual machine being a
    Java virtual machine and the software application
    being a Java application.

3. The mobile device claim 1 or 2, wherein the control
    system requires one digital signature and one sig-
    nature identification for each library of at least one
    of the APIs.

4. The mobile device of any of claims 1 to 3, wherein
    the APIs of the application platform access at least
    one of a cryptographic module, which implements
    cryptographic algorithms, a data store, a proprietary
    data model, and a user interface (UI).

5. The mobile device of any of claims 1 to 4, wherein
    the digital signature is generated using a private sig-
    nature key under the signature scheme, and the ver-
    ification system uses a public signature key to au-
    thenticate the digital signature.

6. The mobile device of claim 5, wherein:

    the digital signature is generated by applying the
    private signature key to a hash of the software
    application under the signature scheme; and
    the verification system authenticates the digital
    signature by generating a hash of the software
    application to obtain a generated hash, applying
    the public signature key to the digital signature
    to obtain a recovered hash, and verifying that
    the generated hash and the recovered hash are
    the same.

7. The mobile device of any of claims 1 to 6, wherein
    at least one of the APIs further comprises:

    a description string that is displayed when the
    software application attempts to access said at
    least one of the APIs.

8. A method of controlling access to application pro-
    gramming interfaces (APIs) on a mobile device, in-
    cluding the step of denying a software application on
    the device access to at least one of the APIs where
    a signature identifier of the such API does not cor-
    respond with a digital signature identification or a
    digital signature provided by the software application
    is not authenticated by a verification system of the
    device.

9. The method of claim 8, comprising the additional
    step of:

    purging the software application from the mobile
    device where the digital signature is not authen-
    ticated.

10. The method of claim 8 or 9, comprising the additional
    step of:

    purging the software application from the mobile
    device where the signature identification does
    not correspond with the signature identifier.

11. The method of any of claims 8 to 10, wherein the
    digital signature and the signature identification are
    generated by a code signing authority.

12. The method of any of claims 9 to 11, wherein a de-
    scription string is displayed to a user when the soft-
    ware application attempts to access said at least one
    of the APIs.

13. The method of any of claims 8 to 12 comprising the
    additional step of:

    displaying a description string that notifies a user
    of the mobile device that the software application
    requires access to the API.

14. The method of claim 13, comprising the additional
    step of:

    receiving a command from the user granting or
    denying the software application access to the
    API.

15. The method of any of claims 8 to 14, comprising the
    additional step of:

    allowing the software application to access the
    API where the digital signature is authenticated.

Figure 1

Figu re 2

Figure 2 flowchart:

Start — 32

Application Y uses Library A — 34

Test Application Y in device simulator with no signature verification. — 36

Application Y forwarded to Code Signing Authority — 38

Application Y reviewed by Code Signing Authority — 40

Accept Code? — 42

No → Rejection Notification to Software Developer — 44

Yes → Code Signing Authority signs Application Y with Digital Signature — 46

Return Application Y to Software Developer with Appended Digital Signature — 48

Application Y loaded on Mobile Device. — 50

End — 52

30

Page 527 of 1415

Figure 3

Figure 3A

Figure 4

Figure 5

Figure 6

Europäisches
Patentamt
European
Patent Office
Office européen
des brevets

EUROPEAN SEARCH REPORT

	DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	ANONYMOUS: "ISO/IEC 7816-4:1995 Information technology -- Identification cards -- Integrated circuit(s) cards with contacts -- Part 4: Interindustry commands for interchange" INTERNATIONAL STANDARD ISO/IEC, vol. 7816-4:1995(E), 1 January 1995 (1995-01-01), pages I-IV,1-46, XP008124701 * page 12 *	1-15	INV. G06F1/00
A	ANONYMOUS: "ISO/IEC 7816-8: IDENTIFICATION CARDS -- INTEGRATED CIRCUIT CARDS - PART 8: COMMANDS FOR SECURITY OPERATIONS" INTERNATIONAL STANDARD ISO/IEC, vol. 7816, no. 8, 25 June 1998 (1998-06-25), XP002610578 Document consists of pages i-iii, 2, 3, 6-13 * table 4 *	1-15	
A	ANONYMOUS: "ISO/IEC 7816-9: IDENTIFICATION CARDS -- INTEGRATED CIRCUIT CARDS - PART 9: COMMANDS FOR CARD MANAGEMENT" INTERNATIONAL STANDARD ISO/IEC, vol. 7816, no. 9, 17 June 1999 (1999-06-17), XP002610579 Document consists of pages i-iv, 9-13, 29-31 * page 9 *	1-15	TECHNICAL FIELDS SEARCHED (IPC) G06F

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	25 November 2010	Kerschbaumer, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

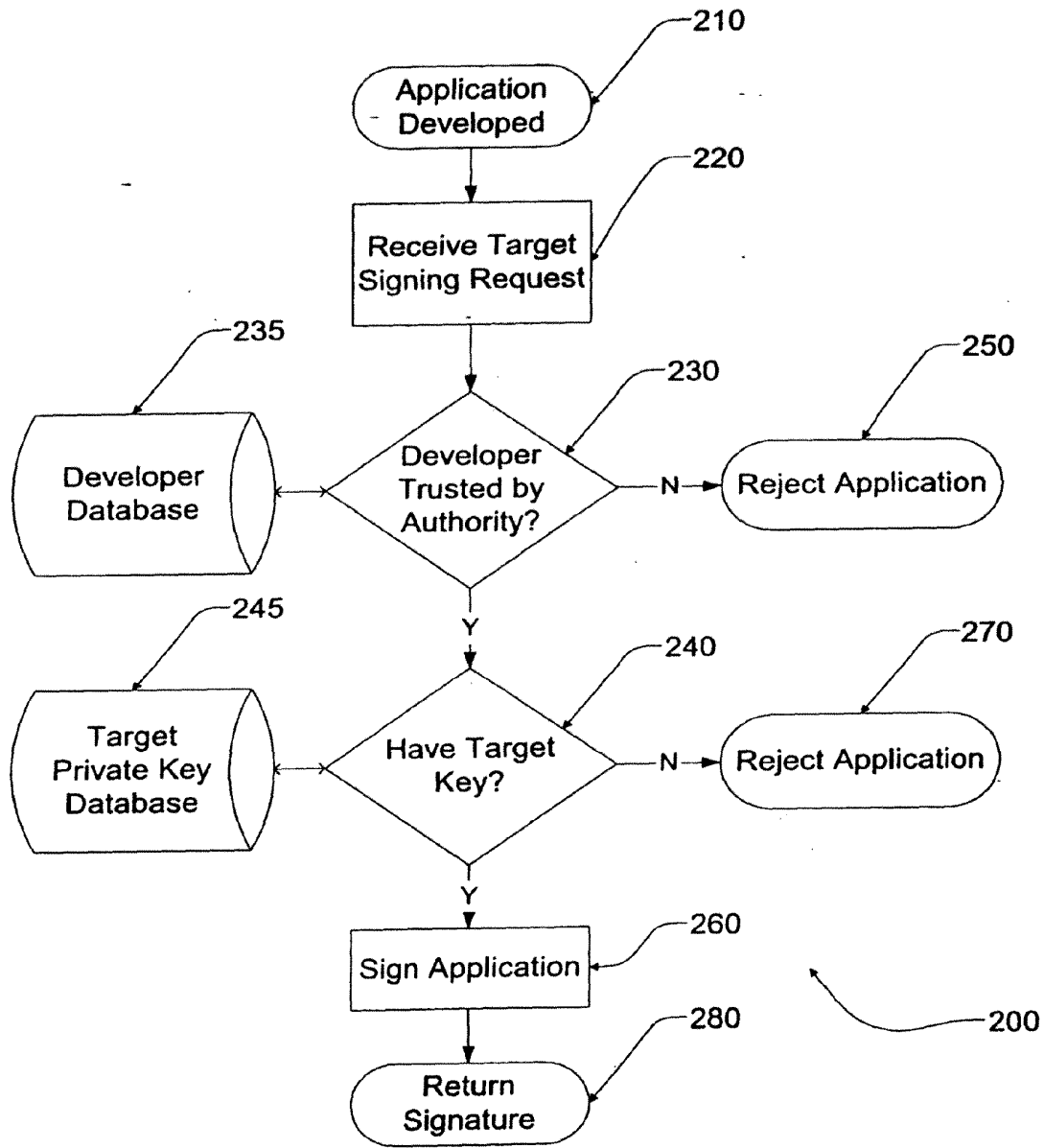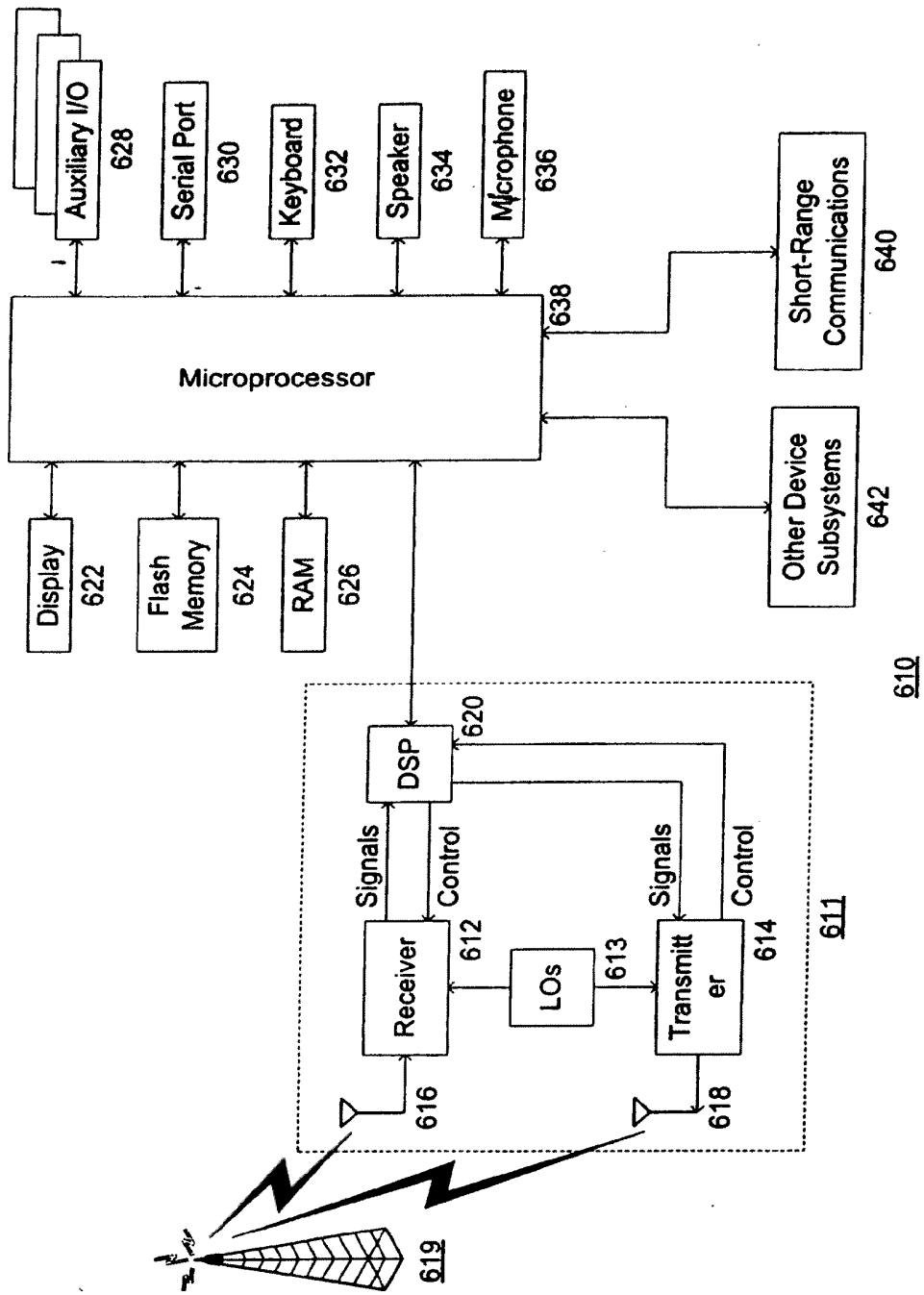| | Europäisches Patentamt
European Patent Office
Office européen des brevets | **EUROPEAN SEARCH REPORT** | Application Number
EP 10 18 3997 |

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	"Excerpts ED - RANKL W; EFFING W" 1 January 1999 (1999-01-01), HANDBUCH DER CHIPKARTEN. AUFBAU - FUNKTIONSWEISE - EINSATZ VON SMART CARDS , MUENCHEN : CARL HANSER VERLAG, DE , XP007908384 ISBN: 978-3-446-21115-5 Document consists of pages 197-203, 261-273, 740-741, 794-797 * pages 269,272; figure 5.39 * -----	1-15	
			TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

3

Place of search	Date of completion of the search	Examiner
Munich	25 November 2010	Kerschbaumer, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

23

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 23415200 P **[0001]**
- US 23535400 P **[0001]**

- US 27066301 P **[0001]**

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

(11)   **EP 2 306 260 A2**

(12)   **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
06.04.2011 Bulletin 2011/14

(21) Application number: **10186296.9**

(22) Date of filing: **20.09.2001**

(51) Int Cl.:
G06F 1/00 (2006.01)

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **21.09.2000 US 234152 P**
26.09.2000 US 235354 P
20.02.2001 US 270663 P

(62) Document number(s) of the earlier application(s) in
accordance with Art. 76 EPC:
05024661.0 / 1 626 324
01973901.0 / 1 320 795

(71) Applicant: **RESEARCH IN MOTION LIMITED**
Waterloo, Ontario N2L 3W8 (CA)

(72) Inventors:
• **Yach, David, P.**
Waterloo Ontario N2K 2N1 (CA)

• **Brown, Michael, S.**
Heidelberg Ontario N0B 1Y0 (CA)
• **Little, Herbert, A.**
Waterloo Ontario N2T 2V8 (CA)

(74) Representative: **MERH-IP**
Matias Erny Reichl Hoffmann
Paul-Heyse-Strasse 29
80336 München (DE)

Remarks:
•Claims filed after the date of filing of the application/
after receipt of the divisional application (Rule 68(4)
EPC).
•This application was filed on 01-10-2010 as a
divisional application to the application mentioned
under INID code 62.

(54)   **SOFTWARE CODE SIGNING SYSTEM AND METHOD**

(57)   A code signing system and method is provided.
The code signing system operates in conjunction with a
signed software application having a digital signature and
includes an application platform, an application program-
ming interface (API), and a virtual machine. The API is
configured to link the software application with the appli-
cation platform. The virtual machine verifies the authen-
ticity of the digital signature in order to control access to
the API by the software application.

Figure 1

EP 2 306 260 A2

Page 536 of 1415

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from and is re-
lated to the following prior applications: "Code Signing
System And Method," U.S. Provisional Application No.
60/234,152, filed Sep. 21, 2000; "Code Signing System
And Method," U.S. Provisional Application No.
60/235,354, filed Sep. 26, 2000; and "Code Signing Sys-
tem And Method," U.S. Provisional Application No.
60/270,663, filed Feb. 20,2001.

BACKGROUND

1. Field of the Invention

[0002] This invention relates generally to the field of
security protocols for software applications. More partic-
ularly, the invention provides a code signing system and
method that is particularly well suited for Java(TM) ap-
plications for mobile communication devices, such as
Personal Digital Assistants, cellular telephones, and
wireless two-way communication devices (collectively
referred to hereinafter as "mobile devices" or simply "de-
vices").

2. Description of the Related Art

[0003] Security protocols involving software code sign-
ing schemes are known. Typically, such security proto-
cols are used to ensure the reliability of software appli-
cations that are downloaded from the Internet. In a typical
software code signing scheme, a digital signature is at-
tached to a software application that identifies the soft-
ware developer. Once the software is downloaded by a
user, the user typically must use his or her judgment to
determine whether or not the software application is re-
liable, based solely on his or her knowledge of the soft-
ware developer's reputation. This type of code signing
scheme does not ensure that a software application writ-
ten by a third party for a mobile device will properly in-
teract with the device's native applications and other re-
sources. Because typical code signing protocols are not
secure and rely solely on the judgment of the user, there
is a serious risk that destructive, "Trojan horse" type soft-
ware applications may be downloaded and installed onto
a mobile device.
[0004] There also remains a need for network opera-
tors to have a system and method to maintain control
over which software applications are activated on mobile
devices.
[0005] There remains a further need in 2.5G and 3G
networks where corporate clients or network operators
would like to control the types of software on the devices
issued to its employees.

SUMMARY

[0006] A code signing system and method is provided.
The code signing system operates in conjunction with a
software application having a digital signature and in-
cludes an application platform, an application program-
ming interface (API), and a virtual machine. The API is
configured to link the software application with the appli-
cation platform. The virtual machine verifies the authen-
ticity of the digital signature in order to control access to
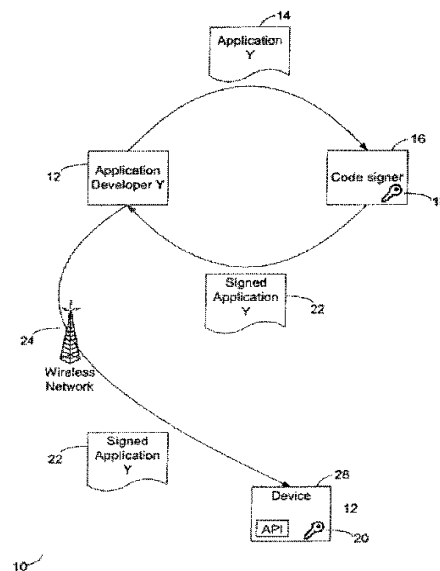the API by the software application.
[0007] A code signing system for operation in conjunc-
tion with a software application having a digital signature,
according to another embodiment of the invention com-
prises an application platform, a plurality of APIs, each
configured to link the software application with a resource
on the application platform, and a virtual machine that
verifies the authenticity of the digital signature in order
to control access to the API by the software application,
wherein the virtual machine verifies the authenticity of
the digital signature in order to control access to the plu-
rality of APIs by the software application.
[0008] According to a further embodiment of the inven-
tion, a method of controlling access to sensitive applica-
tion programming interfaces on a mobile device compris-
es the steps of loading a software application on the mo-
bile device that requires access to a sensitive API, de-
termining whether or not the software application in-
cludes a digital signature associated with the sensitive
API, and if the software application does not include a
digital signature associated with the sensitive API, then
denying the software application access to the sensitive
API.
[0009] In another embodiment of the invention, a meth-
od of controlling access to an application programming
interface (API) on a mobile device by a software appli-
cation created by a software developer comprises the
steps of receiving the software application from the soft-
ware developer, reviewing the software application to de-
termine if it may access the API, if the software applica-
tion may access the API, then appending a digital signa-
ture to the software application, verifying the authenticity
of a digital signature appended to a software application,
and providing access to the API to software applications
for which the appended digital signature is authentic.
[0010] A method of restricting access to a sensitive
API on a mobile device, according to a further embodi-
ment of the invention, comprises the steps of registering
one or more software developers that are trusted to de-
sign software applications which access the sensitive
API, receiving a hash of a software application, deter-
mining if the software application was designed by one
of the registered software developers, and if the software
application was designed by one of the registered soft-
ware developers, then generating a digital signature us-
ing the hash of the software application, wherein the dig-
ital signature may be appended to the software applica-
tion, and the mobile device verifies the authenticity of the

digital signature in order to control access to the sensitive API by the software application.

[0011] In a still further embodiment, a method of restricting access to application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to one or more API, determining whether or not the software application includes a digital signature associated with the mobile device, and if the software application does not include a digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012]

FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention;
FIG. 2 is a flow diagram of the code signing protocol described above with reference to FIG. 1;
FIG. 3 is a block diagram of a code signing system on a mobile device;
FIG. 3A is a block diagram of a code signing system on a plurality of mobile devices;
FIG. 4 is a flow diagram illustrating the operation of the code signing system described above with reference to FIG. 3 and FIG. 3A;
FIG. 5 is a flow diagram illustrating the management of the code signing authorities described with reference to FIG. 3A; and
FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented.

DETAILED DESCRIPTION

[0013] Referring now to the drawing figures, FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention. An application developer 12 creates a software application 14 (application Y) for a mobile device that requires access to one or more sensitive APIs on the mobile device. The software application Y 14 may, for example, be a Java application that operates on a Java virtual machine installed on the mobile device. An API enables the software application Y to interface with an application platform that may include, for example, resources such as the device hardware, operating system and core software and data models. In order to make function calls to or otherwise interact with such device resources, a software application Y must access one or more APIs. APIs can thereby effectively "bridge" a software application and associated device resources. In this description and the appended claims, references to API access should be interpreted to include access of an API in such a way as to allow a software application Y to interact with one or more corresponding device resources. Providing access to any API therefore allows a software application Y to interact with associated device resources, whereas denying access to an API prevents the software application Y from interacting with the associated resources. For example, a database API may communicate with a device file or data storage system, and access to the database API would provide for interaction between a software application Y and the file or data storage system. A user interface (UI) API would communicate with controllers and/or control software for such device components as a screen, a keyboard, and any other device components that provide output to a user or accept input from a user. In a mobile device, a radio API may also be provided as an interface to wireless communication resources such as a transmitter and receiver. Similarly, a cryptographic API may be provided to interact with a crypto module which implements crypto algorithms on a device. These are merely illustrative examples of APIs that may be provided on a device. A device may include any of these example APIs, or different APIs instead of or in addition to those described above.

[0014] Preferably, any API may be classified as sensitive by a mobile device manufacturer, or possibly by an API author, a wireless network operator, a device owner or operator, or some other entity that may be affected by a virus or malicious code in a device software application. For instance, a mobile device manufacturer may classify as sensitive those APIs that interface with cryptographic routines, wireless communication functions, or proprietary data models such as address book or calendar entries. To protect against unauthorized access to these sensitive APIs, the application developer 12 is required to obtain one or more digital signatures from the mobile device manufacturer or other entity that classified any APIs as sensitive, or from a code signing authority 16 acting on behalf of the manufacturer or other entity with an interest in protecting access to sensitive device APIs, and append the signature(s) to the software application Y 14.

[0015] In one embodiment, a digital signature is obtained for each sensitive API or library that includes a sensitive API to which the software application requires access. In some cases, multiple signatures are desirable. This would allow a service provider, company or network operator to restrict some or all software applications loaded or updated onto a particular set of mobile devices. In this multiple-signature scenario, all APIs are restricted and locked until a "global" signature is verified for a software application. For example, a company may wish to prevent its employees from executing any software applications onto their devices without first obtaining permission from a corporate information technology (IT) or computer services department. All such corporate mobile devices may then be configured to require verification of at least a global signature before a software application can be executed. Access to sensitive device APIs and libraries, if any, could then be further restricted, dependent upon verification of respective corresponding digital signatures.

[0016] The binary executable representation of software application Y 14 may be independent of the particular type of mobile device or model of a mobile device. Software application Y 14 may for example be in a write-once-run-anywhere binary format such as is the case with Java software applications. However, it may be desirable to have a digital signature for each mobile device type or model, or alternatively for each mobile device platform or manufacturer. Therefore, software application Y 14 may be submitted to several code signing authorities if software application Y 14 targets several mobile devices.

[0017] Software application Y 14 is sent from the application developer 12 to the code signing authority 16. In the embodiment shown in FIG. 1, the code signing authority 16 reviews the software application Y 14, although as described in further detail below, it is contemplated that the code signing authority 16 may also or instead consider the identity of the application developer 12 to determine whether or not the software application Y 14 should be signed. The code signing authority 16 is preferably one or more representatives from the mobile device manufacturer, the authors of any sensitive APIs, or possibly others that have knowledge of the operation of the sensitive APIs to which the software application needs access.

[0018] If the code signing authority 16 determines that software application Y 14 may access the sensitive API and therefore should be signed, then a signature (not shown) for the software application Y 14 is generated by the code signing authority 16 and appended to the software application Y 14. The signed software application Y 22, comprising the software application Y 14 and the digital signature, is then returned to the application developer 12. The digital signature is preferably a tag that is generated using a private signature key 18 maintained solely by the code signing authority 16. For example, according to one signature scheme, a hash of the software application Y 14 may be generated, using a hashing algorithm such as the Secure Hash Algorithm SHA1, and then used with the private signature key 18 to create the digital signature. In some signature schemes, the private signature key is used to encrypt a hash of information to be signed, such as software application Y 14, whereas in other schemes, the private key may be used in other ways to generate a signature from the information to be signed or a transformed version of the information.

[0019] The signed software application Y 22 may then be sent to a mobile device 28 or downloaded by the mobile device 28 over a wireless network 24. It should be understood, however, that a code signing protocol according to the present invention is not limited to software applications that are downloaded over a wireless network. For instance, in alternative embodiments, the signed software application Y 22 may be downloaded to a personal computer via a computer network and loaded to the mobile device through a serial link, or may be acquired from the application developer 12 in any other

manner and loaded onto the mobile device. Once the signed software application Y 22 is loaded on the mobile device 28, each digital signature is preferably verified with a public signature key 20 before the software application Y 14 is granted access to a sensitive API library. Although the signed software application Y 22 is loaded onto a device, it should be appreciated that the software application that may eventually be executed on the device is the software application Y 14. As described above, the signed software application Y 22 includes the software application Y 14 and one or more appended digital signatures (not shown). When the signatures are verified, the software application Y 14 can be executed on the device and access any APIs for which corresponding signatures have been verified.

[0020] The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority 16, and is preferably installed on the mobile device along with the sensitive API. However, the public key 10 may instead be obtained from a public key repository (not shown), using the device 28 or possibly a personal computer system, and installed on the device 28 as needed. According to one embodiment of a signature scheme, the mobile device 28 calculates a hash of the software application Y 14 in the signed software application Y 22, using the same hashing algorithm as the code signing authority 16, and uses the digital signature and the public signature key 20 to recover the hash calculated by the signing authority 16. The resultant locally calculated hash and the hash recovered from the digital signature are then compared, and if the hashes are the same, the signature is verified. The software application Y 14 can then be executed on the device 28 and access any sensitive APIs for which the corresponding signature(s) have been verified. As described above, the invention is in no way limited to this particular illustrative example signature scheme. Other signature schemes, including further public key signature schemes, may also be used in conjunction with the code signing methods and systems described herein.

[0021] FIG. 2 is a flow diagram 30 of the code signing protocol described above with reference to FIG. 1. The protocol begins at step 32. At step 34, a software developer writes the software application Y for a mobile device that requires access to a sensitive API or library that exposes a sensitive API (API library A). As discussed above, some or all APIs on a mobile device may be classified as sensitive, thus requiring verification of a digital signature for access by any software application such as software application Y. In step 36, application Y is tested by the software developer, preferably using a device simulator in which the digital signature verification function has been disabled. In this manner, the software developer may debug the software application Y before the digital signature is acquired from the code signing authority. Once the software application Y has been written and debugged, it is forwarded to the code signing authority in step 38.

[0022] In steps 40 and 42, the code signing authority reviews the software application Y to determine whether or not it should be given access to the sensitive API, and either accepts or rejects the software application. The code signing authority may apply a number of criteria to determine whether or not to grant the software application access to the sensitive API including, for example, the size of the software application, the device resources accessed by the API, the perceived utility of the software application, the interaction with other software applications, the inclusion of a virus or other destructive code, and whether or not the developer has a contractual obligation or other business arrangement with the mobile device manufacturer. Further details of managing code signing authorities and developers are described below in reference to FIG. 5.

[0023] If the code signing authority accepts the software application Y, then a digital signature, and preferably a signature identification, are appended to the software application Y in step 46. As described above, the digital signature may be generated by using a hash of the software application Y and a private signature key 18. The signature identification is described below with reference to FIGS. 3 and 4. Once the digital signature and signature identification are appended to the software application Y to generate a signed software application, the signed software application Y is returned to the software developer in step 48. The software developer may then license the signed software application Y to be loaded onto a mobile device (step 50). If the code signing authority rejects the software application Y, however, then a rejection notification is preferably sent to the software developer (step 44), and the software application Y will be unable to access any API(s) associated with the signature.

[0024] In an alternative embodiment, the software developer may provide the code signing authority with only a hash of the software application Y, or provide the software application Y in some type of abridged format. If the software application Y is a Java application, then the device independent binary *.class files may be used in the hashing operation, although device dependent files such as *.cod files used by the assignee of the present application may instead be used in hashing or other digital signature operations when software applications are intended for operation on particular devices or device types. By providing only a hash or abridged version of the software application Y, the software developer may have the software application Y signed without revealing proprietary code to the code signing authority. The hash of the software application Y, along with the private signature key 18, may then be used by the code signing authority to generate the digital signature. If an otherwise abridged version of the software application Y is sent to the code signing authority, then the abridged version may similarly be used to generate the digital signature, provided that the abridging scheme or algorithm, like a hashing algorithm, generates different outputs for different inputs. This ensures that every software application will have a different abridged version and thus a different signature that can only be verified when appended to the particular corresponding software application from which the abridged version was generated. Because this embodiment does not enable the code signing authority to thoroughly review the software application for viruses or other destructive code, however, a registration process between the software developer and the code signing authority may also be required. For instance, the code signing authority may agree in advance to provide a trusted software developer access to a limited set of sensitive APIs.

[0025] In still another alternative embodiment, a software application Y may be submitted to more than one signing authority. Each signing authority may for example be responsible for signing software applications for particular sensitive APIs or APIs on a particular model of mobile device or set of mobile devices that supports sensitive APIs required by a software application. A manufacturer, mobile communication network operator, service provider, or corporate client for example may thereby have signing authority over the use of sensitive APIs for their particular mobile device model(s), or the mobile devices operating on a particular network, subscribing to one or more particular services, or distributed to corporate employees. A signed software application may then include a software application and at least one appended digital signature appended from each of the signing authorities. Even though these signing authorities in this example would be generating a signature for the same software application, different signing and signature verification schemes may be associated with the different signing authorities.

[0026] FIG. 3 is a block diagram of a code signing system 60 on a mobile device 62. The system 60 includes a virtual machine 64, a plurality of software applications 66-70, a plurality of API libraries 72-78, and an application platform 80. The application platform 80 preferably includes all of the resources on the mobile device 62 that may be accessed by the software applications 66-70. For instance, the application platform may include device hardware 82, the mobile device's operating system 84, or core software and data models 86. Each API library 72-78 preferably includes a plurality of APIs that interface with a resource available in the application platform. For instance, one API library might include all of the APIs that interface with a calendar program and calendar entry data models. Another API library might include all of the APIs that interface with the transmission circuitry and functions of the mobile device 62. Yet another API library might include all of the APIs capable of interfacing with lower-level services performed by the mobile device's operating system 84. In addition, the plurality of API libraries 72-78 may include both libraries that expose a sensitive API 74 and 78, such as an interface to a cryptographic function, and libraries 72 and 76, that may be accessed without exposing sensitive APIs. Similarly, the

plurality of software applications 66-70 may include both signed software applications 66 and 70 that require access to one or more sensitive APIs, and unsigned software applications such as 68. The virtual machine 64 is preferably an object oriented run-time environment such as Sun Micro System's J2ME(TM) (Java 2 Platform, Micro Edition), which manages the execution of all of the software applications 66-70 operating on the mobile device 62, and links the software applications 66-70 to the various API libraries 72-78.

[0027] Software application Y 70 is an example of a signed software application. Each signed software application preferably includes an actual software application such as software application Y comprising for example software code that can be executed on the application platform 80, one or more signature identifications 94 and one or more corresponding digital signatures 96. Preferably each digital signature 96 and associated signature identification 94 in a signed software application 66 or 70 corresponds to a sensitive API library 74 or 78 to which the software application X or software application Y requires access. The sensitive API library 74 or 78 may include one or more sensitive APIs. In an alternative embodiment, the signed software applications 66 and 70 may include a digital signature 96 for each sensitive API within an API library 74 or 78. The signature identifications 94 may be unique integers or some other means of relating a digital signature 96 to a specific API library 74 or 78, API, application platform 80, or model of mobile device 62.

[0028] API library A 78 is an example of an API library that exposes a sensitive API. Each API library 74 and 78 including a sensitive API should preferably include a description string 88, a public signature key 20, and a signature identifier 92. The signature identifier 92 preferably corresponds to a signature identification 94 in a signed software application 66 or 70, and enables the virtual machine 64 to quickly match a digital signature 96 with an API library 74 or 78. The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority, and is used to verify the authenticity of a digital signature 96. The description string 88 may for example be a textual message that is displayed on the mobile device when a signed software application 66 or 70 is loaded, or alternatively when a software application X or Y attempts to access a sensitive API.

[0029] Operationally, when a signed software application 68-70, respectively including a software application X, Z, or Y, that requires access to a sensitive API library 74 or 78 is loaded onto a mobile device, the virtual machine 64 searches the signed for an appended digital signature 96 associated with the API library 74 or 78. Preferably, the appropriate digital signature 96 is located by the virtual machine 64 by matching the signature identifier 92 in the API library 74 or 78 with a signature identification 94 on the signed software application. If the signed software application includes the appropriate dig-

ital signature 96, then the virtual machine 64 verifies its authenticity using the public signature key 20. Then, once the appropriate digital signature 96 has been located and verified, the description string 88 is preferably displayed on the mobile device before the software application X or Y is executed and accesses the sensitive API. For instance, the description string 88 may display a message stating that "Application Y is attempting to access API Library A," and thereby provide the mobile device user with the final control to grant or deny access to the sensitive API.

[0030] FIG. 3A is a block diagram of a code signing system 61 on a plurality of mobile devices 62E, 62F and 62G. The system 61 includes a plurality of mobile devices each of which only three are illustrated, mobile devices 62E, 62F and 62G. Also shown is a signed software application 70, including a software application Y to which two digital signatures 96E and 96F with corresponding signature identifications 94E and 94F have been appended. In the example system 61, each pair composed of a digital signature and identification, 94E/96E and 94F/96F, corresponds to a model of mobile device 62, API library 78, or associated platform 80. If signature identifications 94E and 94F correspond to different models of mobile device 62, then when a signed software application 70 which includes a software application Y that requires access to a sensitive API library 78 is loaded onto mobile device 62E, the virtual machine 64 searches the signed software application 70 for a digital signature 96E associated with the API library 78 by matching identifier 94E with signature identifier 92. Similarly, when a signed software application 70 including a software application Y that requires access to a sensitive API library 78 is loaded onto a mobile device 62F, the virtual machine 64 in device 62F searches the signed software application 70 for a digital signature 96F associated with the API library 78. However, when a software application Y in a signed software application 70 that requires access to a sensitive API library 78 is loaded onto a mobile device model for which the application developer has not obtained a digital signature, device 62G in the example of FIG. 3A, the virtual machine 64 in the device 64G does not find a digital signature appended to the software application Y and consequently, access to the API library 78 is denied on device 62G. It should be appreciated from the foregoing description that a software application such as software application Y may have multiple device-specific, library-specific, or API-specific signatures or some combination of such signatures appended thereto. Similarly, different signature verification requirements may be configured for the different devices. For example, device 62E may require verification of both a global signature, as well as additional signatures for any sensitive APIs to which a software application, requires access in order for the software application to be executed, whereas device 62F may require verification of only a global signature and device 62G may require verification of signatures only for its sensitive APIs. It should also be ap-

parent that a communication system may include devices (not shown) on which a software application Y received as part of a signed software application such as 70 may execute without any signature verification. Although a signed software application has one or more signatures appended thereto, the software application Y might possibly be executed on some devices without first having any of its signature(s) verified. Signing of a software application preferably does not interfere with its execution on devices in which digital signature verification is not implemented.

[0031] FIG. 4 is a flow diagram 100 illustrating the operation of the code signing system described above with reference to FIGS. 3 and 3A. In step 102, a software application is loaded onto a mobile device. Once the software application is loaded, the device, preferably using a virtual machine, determines whether or not the software application requires access to any API libraries that expose a sensitive API (step 104). If not, then the software application is linked with all of its required API libraries and executed (step 118). If the software application does require access to a sensitive API, however, then the virtual machine verifies that the software application includes a valid digital signature associated any sensitive APIs to which access is required, in steps 106-116.

[0032] In step 106, the virtual machine retrieves the public signature key 20 and signature identifier 92 from the sensitive API library. The signature identifier 92 is then used by the virtual machine in step 108 to determine whether or not the software application has an appended digital signature 96 with a corresponding signature identification 94. If not, then the software application has not been approved for access to the sensitive API by a code signing authority, and the software application is preferably prevented from being executed in step 116. In alternative embodiments, a software application without a proper digital signature 96 may be purged from the mobile device, or may be denied access to the API library exposing the sensitive API but executed to the extent possible without access to the API library. It is also contemplated that a user may be prompted for an input when signature verification fails, thereby providing for user control of such subsequent operations as purging of the software application from the device.

[0033] If a digital signature 96 corresponding to the sensitive API library is appended to the software application and located by the virtual machine, then the virtual machine uses the public key 20 to verify the authenticity of the digital signature 96 in step 110. This step may be performed, for example, by using the signature verification scheme described above or other alternative signature schemes. If the digital signature 96 is not authentic, then the software application is preferably either not executed, purged, or restricted from accessing the sensitive API as described above with reference to step 116. If the digital signature is authentic, however, then the description string 88 is preferably displayed in step 112, warning the mobile device user that the software application re-

quires access to a sensitive API, and possibly prompting the user for authorization to execute or load the software application (step 114). When more than one signature is to be verified for a software application, then the steps 104-110 are preferably repeated for each signature before the user is prompted in step 112. If the mobile device user in step 114 authorizes the software application, then it may be executed and linked to the sensitive API library in step 118.

[0034] FIG. 5 is a flow diagram 200 illustrating the management of the code signing authorities described with reference to FIG. 3A. At step 210, an application developer has developed a new software application which is intended to be executable one or more target device models or types. The target devices may include sets of devices from different manufacturers, sets of device models or types from the same manufacturer, or generally any sets of devices having particular signature and verification requirements. The term "target device" refers to any such set of devices having a common signature requirement. For example, a set of devices requiring verification of a device-specific global signature for execution of all software applications may comprise a target device, and devices that require both a global signature and further signatures for sensitive APIs may be part of more than one target device set. The software application may be written in a device independent manner by using at least one known API, supported on at least one target device with an API library. Preferably, the developed software application is intended to be executable on several target devices, each of which has its own at least one API library.

[0035] At step 220, a code signing authority for one target device receives a target-signing request from the developer. The target signing request includes the software application or a hash of the software application, a developer identifier, as well as at least one target device identifier which identifies the target device for which a signature is being requested. At step 230, the signing authority consults a developer database 235 or other records to determine whether or not to trust developer 220. This determination can be made according to several criteria discussed above, such as whether or not the developer has a contractual obligation or has entered into some other type of business arrangement with a device manufacturer, network operator, service provider, or device manufacturer. If the developer is trusted, then the method proceeds at step 240. However, if the developer is not trusted, then the software application is rejected (250) and not signed by the signing authority. Assuming the developer was trusted, at step 240 the signing authority determines if it has the target private key corresponding to the submitted target identifier by consulting a private key store such as a target private key database 245. If the target private key is found, then a digital signature for the software application is generated at step 260 and the digital signature or a signed software application including the digital signature appended to the software application is returned to the developer at step

280. However, if the target private key is not found at step 240, then the software application is rejected at step 270 and no digital signature is generated for the software application.

[0036] Advantageously, if target signing authorities follow compatible embodiments of the method outlined in FIG. 5, a network of target signing authorities may be established in order to expediently manage code signing authorities and a developer community code signing process providing signed software applications for multiple targets with low likelihood of destructive code.

[0037] Should any destructive or otherwise problematic code be found in a software application or suspected because of behavior exhibited when a software application is executed on a device, then the registration or privileges of the corresponding application developer with any or all signing authorities may also be suspended or revoked, since the digital signature provides an audit trail through which the developer of a problematic software application may be identified. In such an event, devices may be informed of the revocation by being configured to periodically download signature revocation lists, for example. If software applications for which the corresponding digital signatures have been revoked are running on a device, the device may then halt execution of any such software application and possibly purge the software application from its local storage. If preferred, devices may also be configured to re-execute digital signature verifications, for instance periodically or when a new revocation list is downloaded.

[0038] Although a digital signature generated by a signing authority is dependent upon authentication of the application developer and confirmation that the application developer has been properly registered, the digital signature is preferably generated from a hash or otherwise transformed version of the software application and is therefore application-specific. This contrasts with known code signing schemes, in which API access is granted to any software applications arriving from trusted application developers or authors. In the code signing systems and methods described herein, API access is granted on an application-by-application basis and thus can be more strictly controlled or regulated.

[0039] FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented. The mobile communication device 610 is preferably a two-way communication device having at least voice and data communication capabilities. The device preferably has the capability to communicate with other computer systems on the Internet. Depending on the functionality provided by the device, the device may be referred to as a data messaging device, a two-way pager, a cellular telephone with data messaging capabilities, a wireless Internet appliance or a data communication device (with or without telephony capabilities).

[0040] Where the device 610 is enabled for two-way communications, the device will incorporate a communication subsystem 611, including a receiver 612, a transmitter 614, and associated components such as one or more, preferably embedded or internal, antenna elements 616 and 618, local oscillators (LOs) 613, and a processing module such as a digital signal processor (DSP) 620. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem 611 will be dependent upon the communication network in which the device is intended to operate. For example, a device 610 destined for a North American market may include a communication subsystem 611 designed to operate within the Mobitex(TM) mobile communication system or DataTAC(TM) mobile communication system, whereas a device 610 intended for use in Europe may incorporate a General Packet Radio Service (GPRS) communication subsystem 611.

[0041] Network access requirements will also vary depending upon the type of network 919. For example, in the Mobitex and DataTAC networks, mobile devices such as 610 are registered on the network using a unique identification number associated with each device. In GPRS networks however, network access is associated with a subscriber or user of a device 610. A GPRS device therefore requires a subscriber identity module (not shown), commonly referred to as a SIM card, in order to operate on a GPRS network. Without a SIM card, a GPRS device will not be fully functional. Local or non-network communication functions (if any) may be operable, but the device 610 will be unable to carry out any functions involving communications over network 619, other than any legally required operations such as "911" emergency calling.

[0042] When required network registration or activation procedures have been completed, a device 610 may send and receive communication signals over the network 619. Signals received by the antenna 616 through a communication network 619 are input to the receiver 612, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in FIG. 6, analog to digital conversion. Analog to digital conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP 620. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by the DSP 620 and input to the transmitter 614 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission over the communication network 619 via the antenna 618.

[0043] The DSP 620 not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in the receiver 612 and transmitter 614 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 620.

[0044] The device 610 preferably includes a microprocessor 638 which controls the overall operation of the device. Communication functions, including at least data

and voice communications, are performed through the communication subsystem 611. The microprocessor 638 also interacts with further device subsystems or resources such as the display 622, flash memory 624, random access memory (RAM) 626, auxiliary input/output (I/O) subsystems 628, serial port 630, keyboard 632, speaker 634, microphone 636, a short-range communications subsystem 640 and any other device subsystems generally designated as 642. APIs, including sensitive APIs requiring verification of one or more corresponding digital signatures before access is granted, may be provided on the device 610 to interface between software applications and any of the resources shown in FIG. 6.

[0045] Some of the subsystems shown in FIG. 6 perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard 632 and display 622 for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0046] Operating system software used by the microprocessor 638, and possibly APIs to be accessed by software applications, is preferably stored in a persistent store such as flash memory 624, which may instead be a read only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device software applications, or parts thereof, may be temporarily loaded into a volatile store such as RAM 626. It is contemplated that received and transmitted communication signals may also be stored to RAM 626.

[0047] The microprocessor 638, in addition to its operating system functions, preferably enables execution of software applications on the device. A predetermined set of applications which control basic device operations, including at least data and voice communication applications for example, will normally be installed on the device 610 during manufacture. A preferred application that may be loaded onto the device may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the device user such as, but not limited to e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the device to facilitate storage of PIM data items on the device. Such PIM application would preferably have the ability to send and receive data items, via the wireless network. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network, with the device user's corresponding data items stored or associated with a host computer system thereby creating a mirrored host computer on the mobile device with respect to the data items at least. This would be especially advantageous in the case where the host computer system is the mobile device user's office computer system. Further applications, including signed soft-

ware applications as described above, may also be loaded onto the device 610 through the network 619, an auxiliary I/O subsystem 62S, serial port 630, short-range communications subsystem 640 or any other suitable subsystem 642. The device microprocessor 638 may then verify any digital signatures, possibly including both "global" device signatures and API-specific signatures, appended to such a software application before the software application can be executed by the microprocessor 638 and/or access any associated sensitive APIs. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the device 610, through a crypto API and a crypto module which implements crypto algorithms on the device (not shown).

[0048] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem 611 and input to the microprocessor 638, which will preferably further process the received signal for output to the display 622, or alternatively to an auxiliary I/O device 628. A user of device 610 may also compose data items such as email messages for example, using the keyboard 632, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display 622 and possibly an auxiliary I/O device 628. Such composed items may then be transmitted over a communication network through the communication subsystem 611.

[0049] For voice communications, overall operation of the device 610 is substantially similar, except that received signals would preferably be output to a speaker 634 and signals for transmission would be generated by a microphone 636. Alternative voice or audio I/O subsystems such as a voice message recording subsystem may also be implemented on the device 610. Although voice or audio signal output is preferably accomplished primarily through the speaker 634, the display 622 may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information for example.

[0050] The serial port 630 in FIG. 6 would normally be implemented in a personal digital assistant (PDA)-type communication device for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port 630 would enable a user to set preferences through an external device or software application and would extend the capabilities of the device by providing for information or software downloads to the device 610 other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the device through a direct and thus reliable and trusted connection to thereby enable secure device communication.

9

[0051]   A short-range communications subsystem 640 is a further optional component which may provide for communication between the device 624 and different systems or devices, which need not necessarily be similar devices. For example, the subsystem 640 may include an infrared device and associated circuits and components or a Bluetooth(TM) communication module to provide for communication with similarly-enabled systems and devices.

[0052]   The embodiments described herein are examples of structures, systems or methods having elements corresponding to the elements of the invention recited in the claims. This written description may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements of the invention recited in the claims. The intended scope of the invention thus includes other structures, systems or methods that do not differ from the literal language of the claims, and further includes other structures, systems or methods with insubstantial differences from the literal language of the claims.

[0053]   For example, when a software application is rejected at step 250 in the method shown in FIG. 5, the signing authority may request that the developer sign a contract or enter into a business relationship with a device manufacturer or other entity on whose behalf the signing authority acts. Similarly, if a software application is rejected at step 270, authority to sign the software application may be delegated to a different signing authority. The signing of a software application following delegation of signing of the software application to the different authority can proceed substantially as shown in FIG. 5, wherein the target signing authority that received the original request from the trusted developer at step 220 requests that the software application be signed by the different signing authority on behalf of the trusted developer from the target signing authority. Once a trust relationship has been established between code signing authorities, target private code signing keys could be shared between code signing authorities to improve performance of the method at step 240, or a device may be configured to validate digital signatures from either of the trusted signing authorities.

[0054]   In addition, although described primarily in the context of software applications, code signing systems and methods according to the present invention may also be applied to other device-related components, including but in no way limited to, commands and associated command arguments, and libraries configured to interface with device resources. Such commands and libraries may be sent to mobile devices by device manufacturers, device owners, network operators, service providers, software application developers and the like. It would be desirable to control the execution of any command that may affect device operation, such as a command to change a device identification code or wireless communication network address for example, by requiring verification of one or more digital signatures before a command can be executed on a device, in accordance with the code signing systems and methods described and claimed herein.

[0055]   As has been described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; an application programming interface (API) configured to link the software application with the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0056]   The virtual machine may deny the software application access to the API if the digital signature is not authentic. The virtual machine may purge the software application if the digital signature is not authentic. The code signing system may be installed on a mobile device. The digital signature may be generated by a code signing authority.

[0057]   The code signing system may further comprise a plurality of API libraries each including a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

[0058]   One or more of the plurality of API libraries may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive API libraries by the software application. The software application may include a unique digital signature for each sensitive API library. The software application may include a signature identification for each unique digital signature; each sensitive API library may include a signature identifier; and the virtual machine may compare the signature identification and the signature identifier to match the unique digital signatures with sensitive API libraries.

[0059]   The digital signature may be generated using a private signature key, and the virtual machine may use a public signature key to verify the authenticity of the digital signature. The digital signature may be generated by applying the private signature key to a hash of the software application; and the virtual machine may verify the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

[0060]   The API may further comprise a description string that is displayed by the mobile device when the software application attempts to access the API. The application platform may comprise an operating system. The application platform may comprise one or more core functions of a mobile device. The application platform may comprise hardware on a mobile device. The hardware may comprise a subscriber identity module (SIM) card. The software application may be a Java application for a mobile device. The API may interface with a cryptographic routine on the application platform. The API may interface with a proprietary data model on the application platform. The virtual machine may be a Java

virtual machine installed on a mobile device.

[0061] As also described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; a plurality of application programming interfaces (APIs), each configured to link the software application with a resource on the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0062] The plurality of APIs may be included in an API library. One or more of the plurality of APIs may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive APIs. For operation in conjunction with a plurality of software applications, one or more of the plurality of software applications may have a digital signature, and the virtual machine may verify the authenticity of the digital signature of each of the one or more of the plurality of software applications in order to control access to the sensitive APIs by each of the plurality of software applications. The resource on the application platform may comprise a wireless communication system. The resource on the application platform may comprise a cryptographic module which implements cryptographic algorithms. The resource on the application platform may comprise a data store. The resource on the application platform may comprise a user interface (UI).

[0063] As has also been described, a method of controlling access to sensitive application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to a sensitive application programming interface (API); determining whether or not the software application includes a digital signature associated with the sensitive API; and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0064] The method my comprise the additional step of: if the software application does not include a digital signature associated with the sensitive API, then purging the software application from the mobile device. The digital signature may be generated by a code signing authority. The method may comprise the additional steps of: if the software application includes a digital signature associated with the sensitive API, then verifying the authenticity of the digital signature; and if the digital signature is not authentic, then denying the software application access to the sensitive API. The method may further comprise the additional step of: if the digital signature is not authentic, then purging the software application from the mobile device. The digital signature may be generated by applying a private signature key to a hash of the software application, and the step of verifying the authenticity of the digital signature may be performed by a meth-

od comprising the steps of: storing a public signature key that corresponds to the private signature key on the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash. The digital signature may be generated by calculating a hash of the software application and applying the private signature key. The method may comprise the additional step of: displaying a description string that notifies a user of the mobile device that the software application requires access to the sensitive API. The method may further comprise the additional step of: receiving a command from the user granting or denying the software application access to the sensitive API.

[0065] Further has been described a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer, comprising the steps of: receiving the software application from the software developer; reviewing the software application to determine if it may access the API; if the software application may access the API, then appending a digital signature to the software application; verifying the authenticity of a digital signature appended to a software application; and providing access to the API to software applications for which the appended digital signature is authentic.

[0066] The step of reviewing the software application may be performed by a code signing authority. The step of appending the digital signature to the software application may be performed by a method comprising the steps of: calculating a hash of the software application; and applying a signature key to the hash of the software application to generate the digital signature. The hash of the software application may be calculated using the Secure Hash Algorithm (SHA1). The step of verifying the authenticity of a digital signature may comprise the steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; and determining if the digital signature is authentic by comparing the calculated hash with the recovered hash. The method may further comprise the step of, if the digital signature is not authentic, then denying the software application access to the API. The signature key may be a private signature key and the corresponding signature key is a public signature key.

[0067] Also has been described, a method of controlling access to a sensitive application programming interface (API) on a mobile device, comprising the steps of: registering one or more software developers that are trusted to design software applications which access the sensitive API; receiving a hash of a software application; determining if the software application was designed by one of the registered software developers; and if the software application was designed by one of the registered software developers, then generating a digital signature

using the hash of the software application, wherein the digital signature may be appended to the software application; and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0068] The step of generating the digital signature may be performed by a code signing authority. The step of generating the digital signature may be performed by applying a signature key to the hash of the software application. The mobile device may verify the authenticity of the digital signature by performing the additional steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; determining if the digital signature is authentic by comparing the calculated hash with the recovered hash; and if the digital signature is not authentic, then denying the software application access to the sensitive API.

[0069] As has been described, a method of restricting access to application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to one or more application programming interface (API); determining whether or not the software application includes an authentic digital signature associated with the mobile device; and if the software application does not include an authentic digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

[0070] The method may comprise the additional step of: if the software application does not include an authentic digital signature associated with the mobile device, then purging the software application from the mobile device. The software application may include a plurality of digital signatures. The plurality of digital signatures may include digital signatures respectively associated with different types of mobile devices.

[0071] Each of the plurality of digital signatures may be generated by a respective corresponding code signing authority. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the additional steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature. The one or more APIs may include one or more APIs classified as sensitive, and the method may further comprise the steps of, for each sensitive API: determining whether or not the software application includes an authentic digital signature associated with the sensitive API; and if the software application does not include an authentic digital signature associated with the sensitive API, then denying the software application access to the sensitive API. Each of the plurality of digital signatures may be generated by its corresponding code signing authority by applying a respective private signa-

ture key associated with the code signing authority to a hash of the software application. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature, wherein the step of verifying the authenticity of the digital signature is performed by a method comprising the steps of: storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the signature associated with the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash.

[0072] In the following, a summary of some aspects of the application is disclosed:

> 1. A code signing system for operation in conjunction with a software application (66) having a digital signature (96) and a signature identification (94), where the digital signature (96) is associated with the signature identification (94), comprising:
>
> - an application platform;
> - an application programming interface (API) having an associated signature identifier (92), the API is configured to link the software application (66) with the application platform; and
> - a virtual machine (64) that verifies the authenticity of the digital signature (96) in order to control access to the API by the software application (66) where the signature identifier (92) corresponds to the signature identification (94).
>
> 2. The code signing system of aspect 1, wherein
>
> > (i) the virtual machine (64) may deny the software application (66) access to the API if the digital signature (96) is not authenticated, or
> > (ii) wherein the virtual machine (64) may purge the software application (66) if the digital signature (96) is not authenticated,
>
> 3. The code signing system of aspect 1 or 2, wherein
>
> > (iii) the code signing system may be installed on a mobile device (62), or
> > (iv) wherein the digital signature (96) may be generated by a code signing authority.
>
> 4. The code signing system of any of aspects 1 to 3, optionally further comprising:
>
> - a plurality of API libraries, each of the plurality

of API libraries may include a plurality of APIs, wherein the virtual machine (64) may control access to the plurality of API libraries by the software application (66).

5. The code signing system of any of aspects 1 to 4,

- wherein at least one of the plurality of API libraries may be classified as sensitive;
- wherein access to a sensitive API library may require a digital signature (96) associated with a signature identification (94) where the signature identification (94) corresponds to a signature identifier (92) associated with the sensitive API library;
- wherein the software application (66) may include at least one digital signature (96) and at least one associated signature identification (94) for accessing sensitive API libraries; and
- wherein the virtual machine (64) may authenticate the software application (66) for accessing the sensitive API library by verifying the one digital signature (96) included in the software application (66) that has a signature identification (94) corresponding to the signature identifier (92) of the sensitive API library.

6. The code signing system of any of aspects 1 to 5, wherein the digital signature (96) may be generated using a private signature key, and the virtual machine (64) may use a public signature key to verify the authenticity of the digital signature (96).

7. The code signing system of aspect 6, wherein:

- the digital signature (96) may be generated by applying the private signature key to a hash of the software application (66); and
- the virtual machine (64) may verify the authenticity of the digital signature (96) by generating a hash of the software application (66) to obtain a generated hash, applying the public signature key to the digital signature (96) to obtain a recovered hash, and comparing the generated hash with the recovered hash.

8. The code signing system of aspect 3, wherein the API optionally further comprises:

- a description string (88) that is displayed by the mobile device (62) when the software application (66) attempts to access the API.

9. The code signing system of any of aspects 1 to 7, wherein the application platform

(i) may comprise an operating system (84), or
(ii) may comprise one or more core functions

(86) of a mobile device (62), or
(iii) may comprise hardware (82) on a mobile device (62).

10. The code signing system of aspect 9, wherein the hardware (82) may comprise a subscriber identity module (SIM) card.

11. The code signing system of any of aspects 1 to 10, wherein the software application (66) may be a Java application for a mobile device (62).

12. The code signing system of any of aspects 1 to 11, wherein

(i) the API may interface with a cryptographic routine on the application platform, or wherein
(ii) the API may interface with a proprietary data model on the application platform.

13. The code signing system of any of aspects 1 to 12, wherein the virtual machine (64) may be a Java virtual machine installed on a mobile device (62).

14. A method of controlling access to sensitive application programming interfaces on a mobile device (62), comprising the steps of:

- loading a software application (66) on the mobile device (62) that requires access to a sensitive application programming interface (API) having a signature identifier (92);
- determining whether the software application (66) includes a digital signatur (96) and a signature identification (94); and
- denying the software application (66) access to the sensitive API where the signature identification (94) does not correspond with the signature identifier (92).

15. The method of aspect 14, optionally comprising the additional step of:

- purging the software application (66) from the mobile device (62) where the signature identification (94) does not correspond with the signature identifier (92).

16. The method of aspect 14 or aspect 15, wherein the digital signature (96) and the signature identification (94) may be generated by a code signing authority.

17. The method of any of aspects 14 to 16, optionally comprising the additional steps of:

- verifying the authenticity of the digital signature (96) where the signature identification (94) cor-

responds with the signature identifier (92); and

- denying the software application (66) access to the sensitive API where the digital signature (96) is not authenticated.

18. The method of aspect 17, optionally comprising the additional step of:

- purging the software application (66) from the mobile device (62) where the digital signature (96) is not authenticated.

19. The method of aspect 17, wherein the digital signature (96) may be generated by applying a private signature key to a hash of the software application (66), and wherein the step of verifying the authenticity of the digital signature (96) may be performed by a method comprising the steps of:

- storing a public signature key that corresponds to the private signature key on the mobile device (62);
- generating a hash of the software application (66) to obtain a generated hash;
- applying the public signature key to the digital signature (96) to obtain a recovered hash; and
- comparing the generated hash with the recovered hash.

20. The method of aspect 19, wherein the digital signature (96) may be generated by calculating a hash of the software application (66) and applying the private signature key.

21. The method of any of aspects 14 to 20, optionally comprising the additional step of:

- displaying a description string (88) that notifies a user of the mobile device (62) that the software application (66) requires access to the sensitive API.

22. The method of aspect 21, optionally comprising the additional step of:

- receiving a command from the user granting or denying the software application (66) access to the sensitive API.

23. A mobile device for a mobile device comprising:

- an application platform having application programming interfaces (APIs);
- a verification system for authenticating digital signatures (96) and signature identifications (94) provided by the respective software applications (66) to access the APIs; and
- a control system for allowing a software appli-

cation (66) to access at least one of the APIs where a digital signature (96) provided by the software application (66) is authenticated by the verification system;

- wherein a code signing authority provides digital signatures (96) and signature identifications (94) to software applications (66) that require access to at least one of the APIs such that the digital signature (96) for the software application (66) is generated according to a signature scheme of a signature identification (94), and wherein the signature identifications (94) provided to the software applications (66) comprise those signature identifications (94) that are substantially only authorized to allow access on the subset of the plurality of mobile devices (62).

24. The mobile device of aspect 23, wherein a virtual machine (64) may comprise the verification system and the control system, preferably the virtual machine (64) being a Java virtual machine and the software application being a Java application.

25. The mobile device of aspect 23 or 24, wherein the control system may require one digital signature (96) and one signature identification (94) for each library of at least one of the APIs.

26. The mobile device of any of aspects 23 to 25, wherein the APIs of the application platform may access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

27. The mobile device of any of aspects 23 to 26, wherein the digital signature (96) may be generated using a private signature key under the signature scheme, and the verification system may use a public signature key to authenticate the digital signature.

28. The mobile device of aspect 27, wherein:

- the digital signature (96) may be generated by applying the private signature key to a hash of the software application (66) under the signature scheme; and
- the verification system may authenticate the digital signature (96) by generating a hash of the software application (66) to obtain a generated hash, applying the public signature key to the digital signature (96) to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

29. The mobile device of any of aspects 23 to 28, wherein at least one of the APIs optionally further comprises:

- a description string (88) that is displayed to a user when the software application (66) attempts to access said at least one of the APIs.

Claims

1. A method of controlling access by code to resources of a device (62), the method comprising:

    determining whether the code includes an authentic digital signature (96) corresponding to one or more application programming interfaces (APIs) of the device; and
    controlling access by the code to the one or more APIs depending on whether the code includes an authentic digital signature (96).

2. The method of claim 1, further comprising:

    purging the code from the device (62) if the code does not include an authentic digital signature (96).

3. The method of any preceding claim, wherein controlling access by the code comprises:

    denying the code access to the one or more APIs if the code does not include an authentic digital signature (96).

4. The method of any preceding claim, wherein controlling access by the code comprises:

    granting the code access to the one or more APIs if the code includes an authentic digital signature (96).

5. The method of any preceding claim,
    wherein the code includes a plurality of digital signatures (96); and
    wherein the plurality of digital signatures (96) includes digital signatures corresponding to different APIs.

6. The method of any preceding claim,
    wherein the code includes a plurality of digital signatures (96); and
    wherein the plurality of digital signatures (96) includes digital signatures associated with different types of devices.

7. The method of claim 5 or 6, wherein each of the plurality of digital signatures (96) was generated by a respective corresponding code signing authority.

8. The method of claim 7, wherein each of the plurality of digital signatures (96) was generated by its cor-

responding code signing authority by applying a respective private key associated with the code signing authority to a hash of the code.

9. The method of any preceding claim, wherein determining whether the code includes an authentic digital signature (96) comprises:

    generating a hash of the code to obtain a generated hash;
    applying a public key to a digital signature (96) included in the code to obtain a recovered hash, wherein the public key corresponds to a private key associated with a code signing authority that generated the digital signature (96); and
    comparing the generated hash with the recovered hash.

10. The method of any preceding claim, wherein determining whether the code includes an authentic digital signature (96) comprises:

    determining whether the code includes an authentic global signature.

11. The method of any preceding claim, wherein the code comprises any of the following: a software application, an update to a software application, a command, a command argument, or a library.

12. The method of any preceding claim, further comprising:

    displaying a message if the code attempts to access at least one of the APIs.

13. The method of any preceding claim, further comprising:

    receiving a user command granting or denying the code access to at least one of the APIs.

14. A device for performing the method of any one of claims 1 - 13.

15. Code for performing the method of any one of claims 1 - 13.
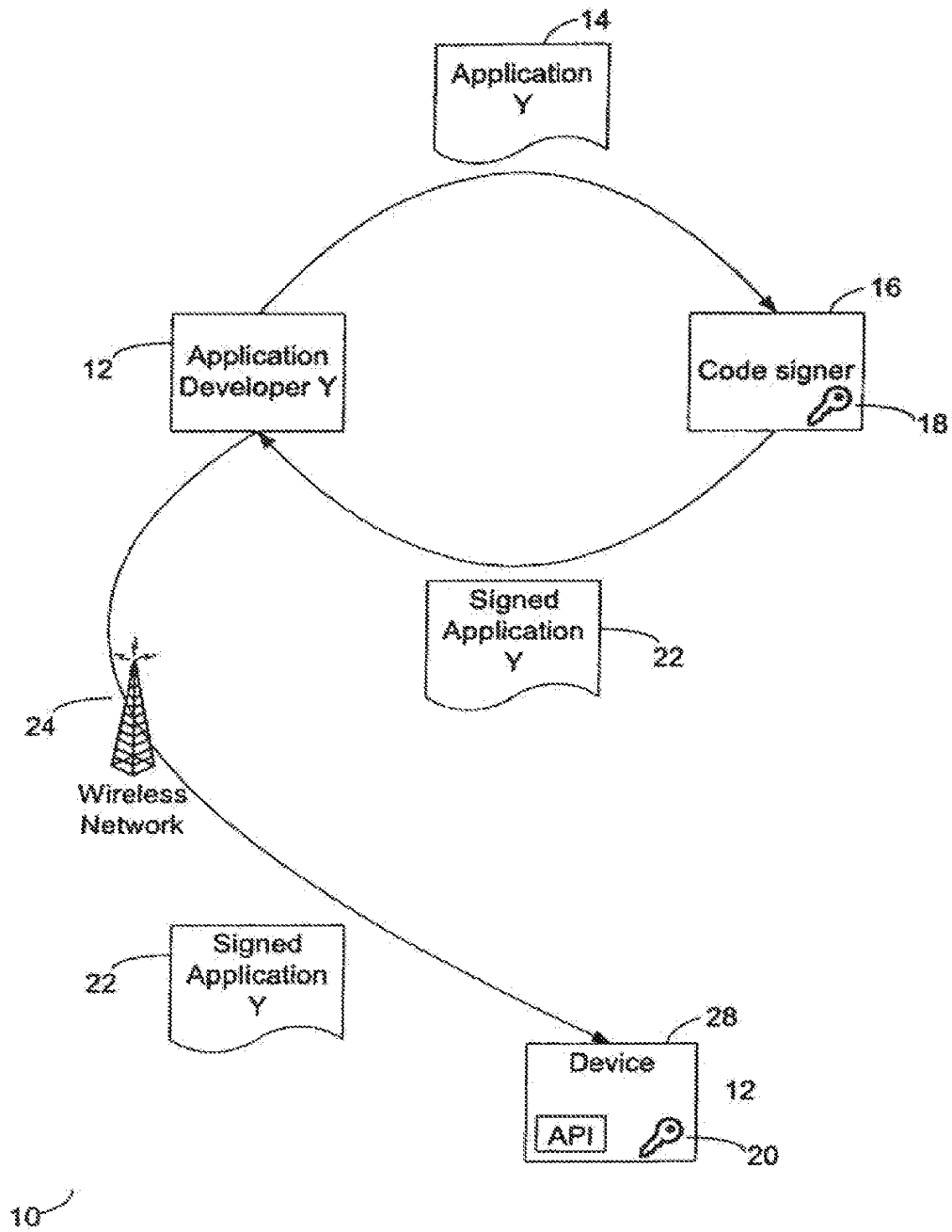
Figure 1
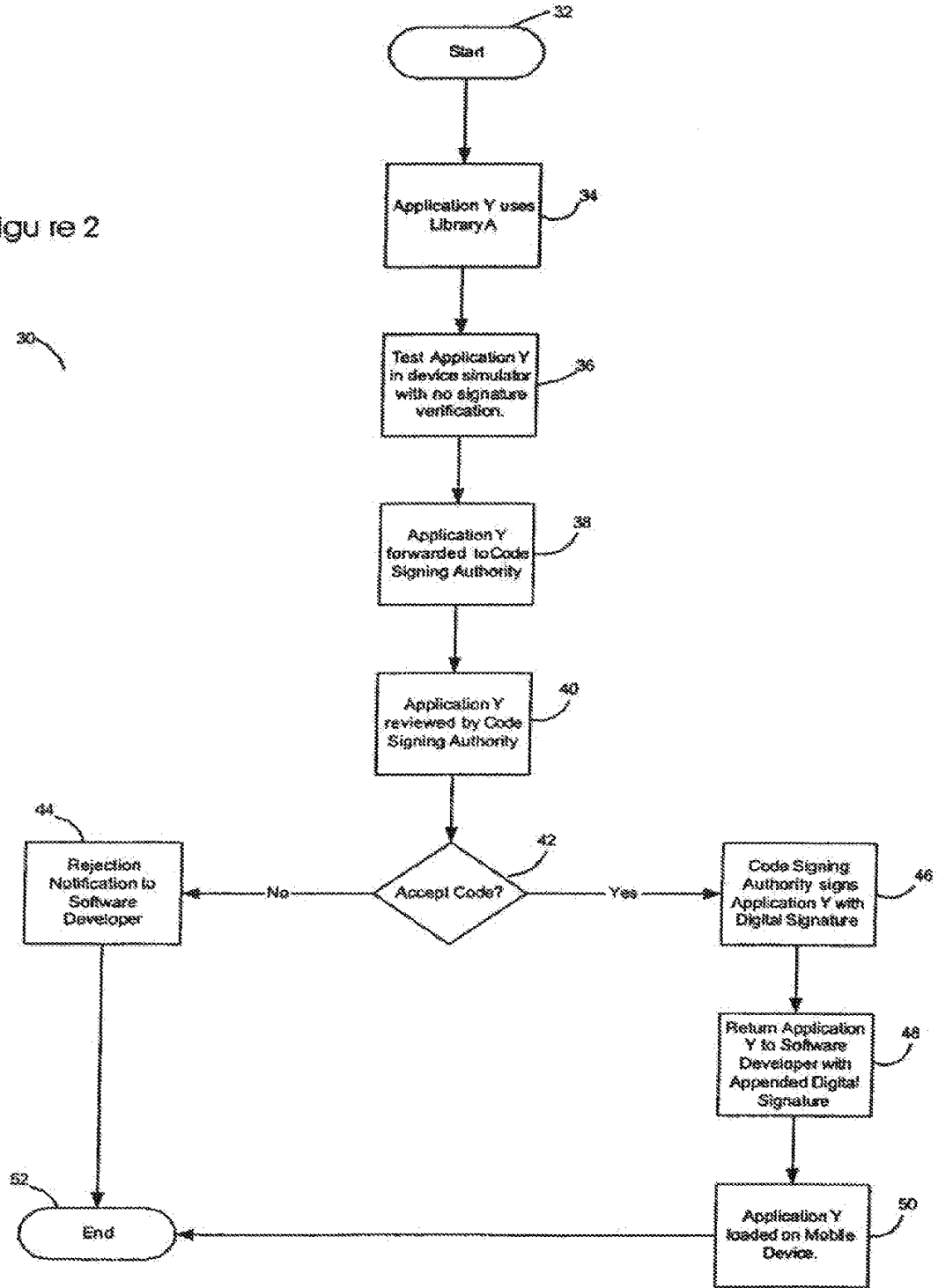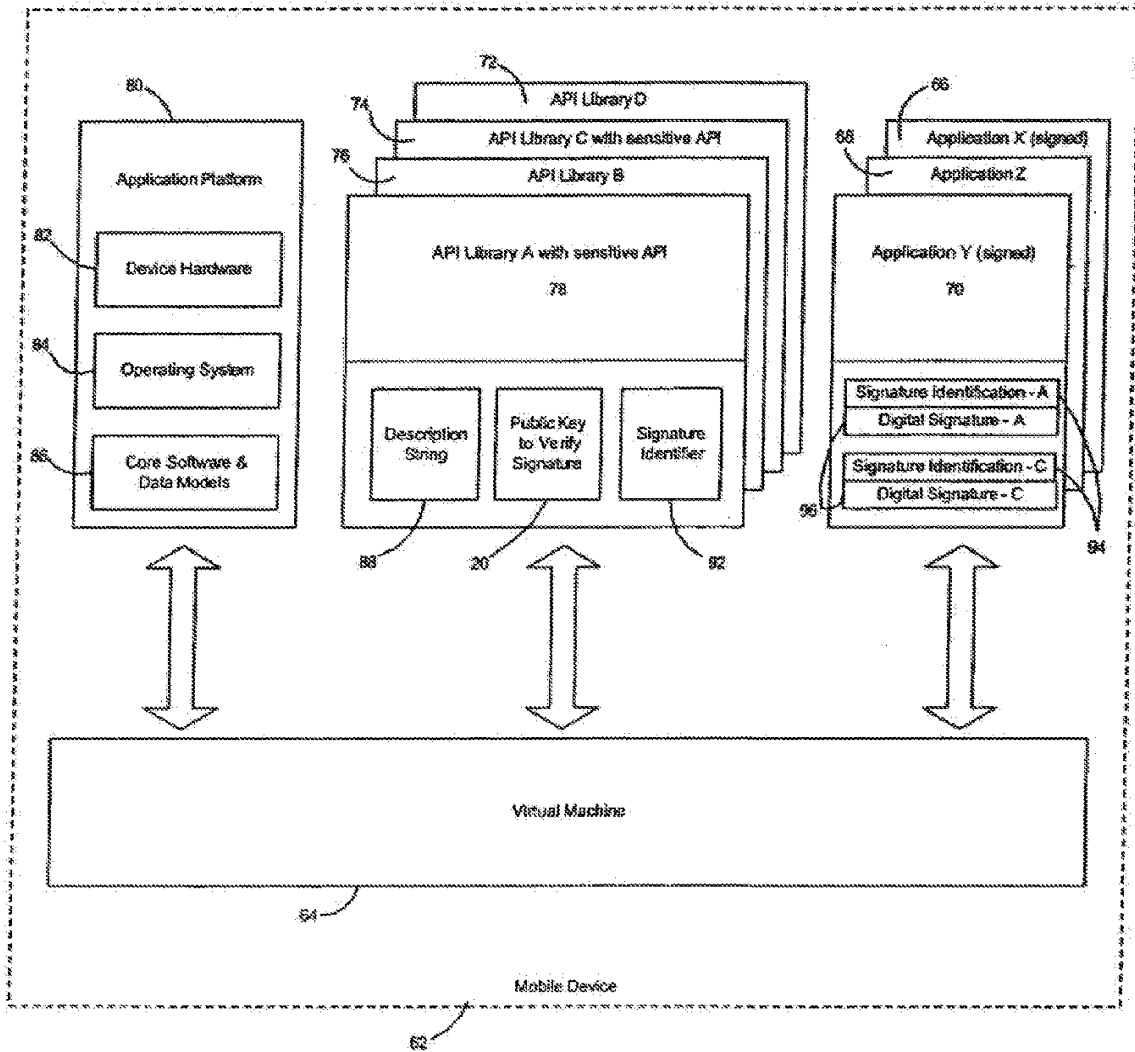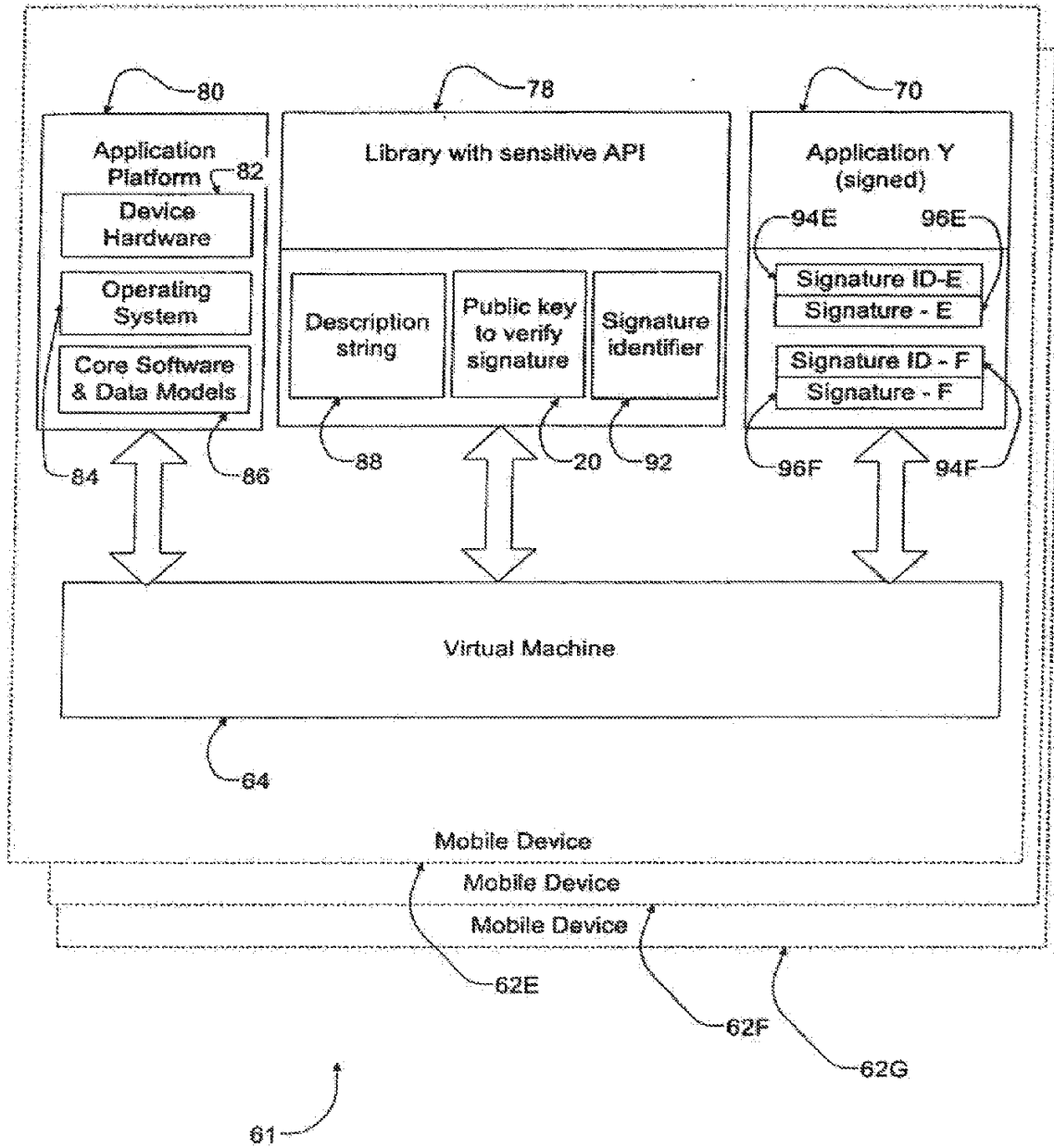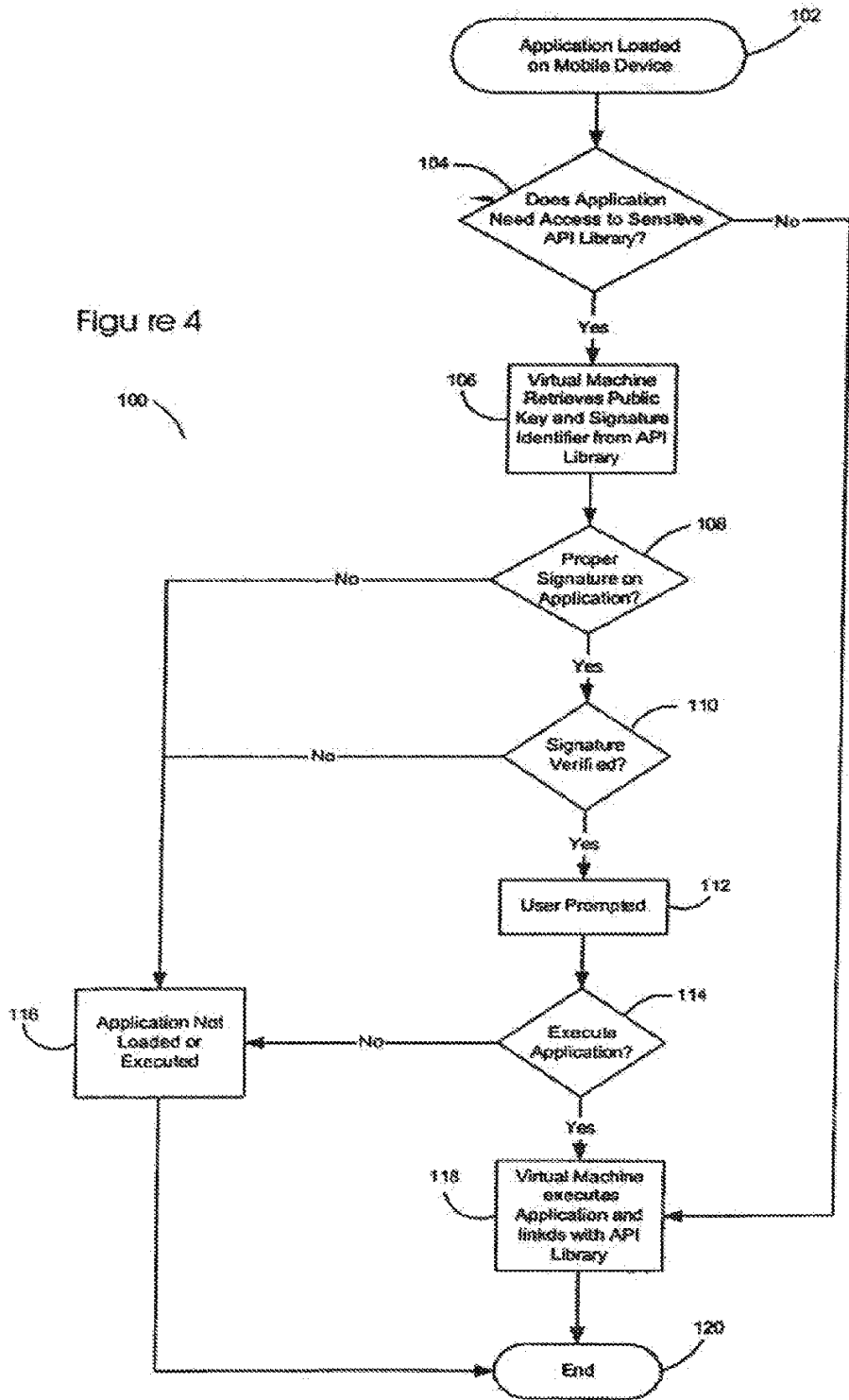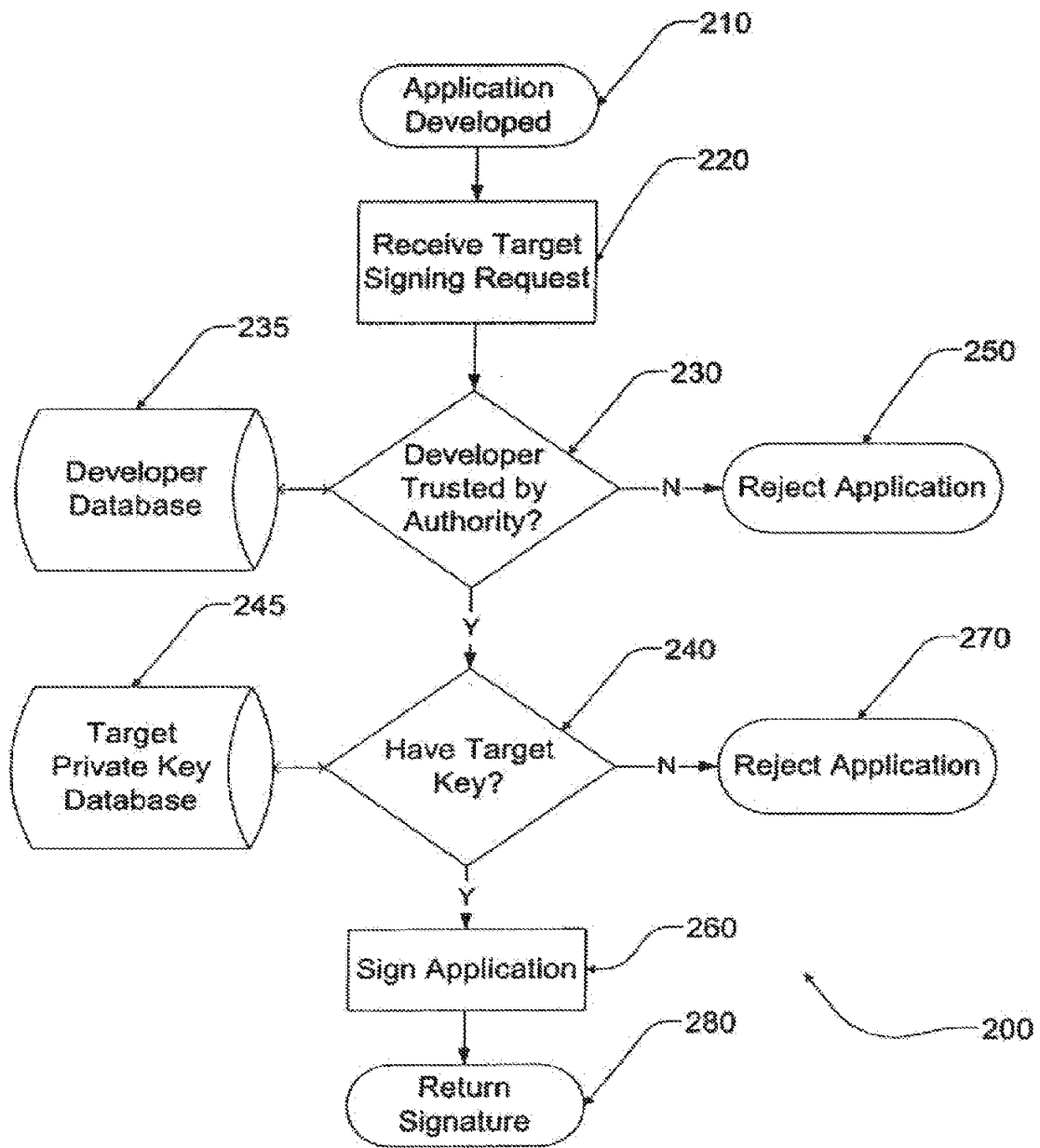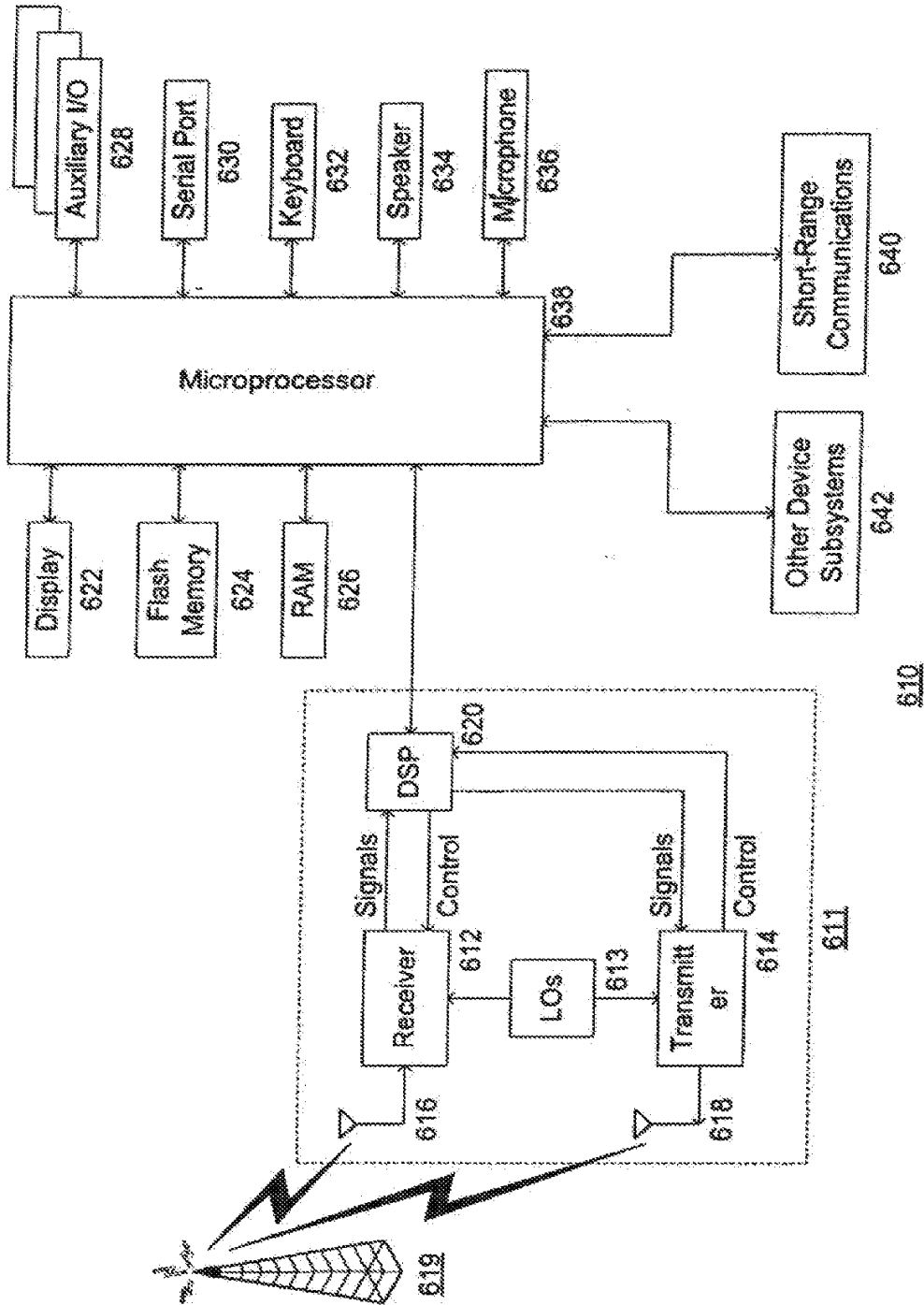
16

Figure 2

Page 552 of 1415

Figure 3

Figure 3A

Figu re 4

102 — Application Loaded on Mobile Device

104 — Does Application Need Access to Sensitive API Library? — No

100

Yes

106 — Virtual Machine Retrieves Public Key and Signature Identifier from API Library

108 — Proper Signature on Application? — No

Yes

110 — Signature Verified? — No

Yes

112 — User Prompted

114 — Execute Application? — No

116 — Application Not Loaded or Executed

Yes

118 — Virtual Machine executes Application and linkds with API Library

120 — End

Figure 5

Figure 6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 23415200 P **[0001]**
- US 23535400 P **[0001]**
- US 27066301 P **[0001]**

23

(19)

Europäisches
Patentamt
European
Patent Office
Office européen
des brevets

(11) EP 1 626 325 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
01.09.2010 Bulletin 2010/35

(51) Int Cl.:
G06F 1/00 (2006.01)

(21) Application number: **05024662.8**

(22) Date of filing: **20.09.2001**

(54) **SOFTWARE CODE SIGNING SYSTEM AND METHOD**

SYSTEM UND VERFAHREN ZUM UNTERSCHREIBEN EINES SOFTWARE-KODES

SYSTEME ET PROCEDE DE SIGNATURE PAR CODE

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**

(30) Priority: **21.09.2000 US 234152 P
26.09.2000 US 235354 P
20.02.2001 US 270663 P**

(43) Date of publication of application:
15.02.2006 Bulletin 2006/07

(62) Document number(s) of the earlier application(s) in
accordance with Art. 76 EPC:
01973901.0 / 1 320 795

(73) Proprietor: **RESEARCH IN MOTION LIMITED
Waterloo, Ontario N2L 3W8 (CA)**

(72) Inventors:
• **Yach, David P.
Waterloo,
Ontario N2K 2N1 (CA)**

• **Brown, Michael S.
Heidelberg,
Ontario N0B 1Y0 (CA)**
• **Little, Herbert A.
Waterloo,
Ontario N2T 2V8 (CA)**

(74) Representative: **MERH-IP
Matias Erny Reichl Hoffmann
Paul-Heyse-Strasse 29
80336 München (DE)**

(56) References cited:
**US-A- 5 625 690   US-A- 5 978 484
US-A- 6 067 582**

• **"Excerpts ED - RANKL W; EFFING W"
HANDBUCH DER CHIPKARTEN. AUFBAU -
FUNKTIONSWEISE - EINSATZ VON SMART
CARDS, MUENCHEN : CARL HANSER VERLAG,
DE, 1 January 1999 (1999-01-01), pages
197-203,261, XP007908384 ISBN:
978-3-446-21115-5**

EP 1 626 325 B1

Description

BACKGROUND

1. Field of the Invention

[0001] This invention relates generally to the field of security protocols for software applications. More particularly, the invention provides a code signing system and method that is particularly well suited for Java(TM) applications for mobile communication devices, such as Personal Digital Assistants, cellular telephones, and wireless two-way communication devices (collectively referred to hereinafter as "mobile devices" or simply "devices").

2. Description of the Related Art

[0002] Security protocols involving software code signing schemes are known. Typically, such security protocols are used to ensure the reliability of software applications that are downloaded from the Internet. In a typical software code signing scheme, a digital signature is attached to a software application that identifies the software developer. Once the software is downloaded by a user, the user typically must use his or her judgment to determine whether or not the software application is reliable, based solely on his or her knowledge of the software developer's reputation. This type of code signing scheme does not ensure that a software application written by a third party for a mobile device will properly interact with the device's native applications and other resources. Because typical code signing protocols are not secure and rely solely on the judgment of the user, there is a serious risk that destructive, "Trojan horse" type software applications may be downloaded and installed onto a mobile device.

[0003] The disclosure "Handbuch der Chipkarten", W.Rankl/W.Effing, 3. edition, 1999, describes access control via methods to securely load muliple applets in a javacard framework System.

[0004] There also remains a need for network operators to have a system and method to maintain control over which software applications are activated on mobile devices.

[0005] There remains a further need in 2.5G and 3G networks where corporate clients or network operators would like to control the types of software on the devices issued to its employees.

SUMMARY

[0006] A code signing system and method is provided. The code signing system operates in conjunction with a software application having a digital signature and includes an application platform, an application programming interface (API), and a virtual machine. The API is configured to link the software application with the appli-

cation platform. The virtual machine verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0007] A code signing system for operation in conjunction with a software application having a digital signature, according to another embodiment of the invention comprises an application platform, a plurality of APIs, each configured to link the software application with a resource on the application platform, and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0008] According to a further embodiment of the invention, a method of controlling access to sensitive application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to a sensitive API, determining whether or not the software application includes a digital signature associated with the sensitive API, and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0009] In another embodiment of the invention, a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer comprises the steps of receiving the software application from the software developer, reviewing the software application to determine if it may access the API, if the software application may access the API, then appending a digital signature to the software application, verifying the authenticity of a digital signature appended to a software application, and providing access to the API to software applications for which the appended digital signature is authentic.

[0010] A method of restricting access to a sensitive API on a mobile device, according to a further embodiment of the invention, comprises the steps of registering one or more software developers that are trusted to design software applications which access the sensitive API, receiving a hash of a software application, determining if the software application was designed by one of the registered software developers, and if the software application was designed by one of the registered software developers, then generating a digital signature using the hash of the software application, wherein the digital signature may be appended to the software application, and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0011] In a still further embodiment, a method of restricting access to application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to one or more API, determining whether or not the software application includes a digital signature associ-

ated with the mobile device, and if the software application does not include a digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012]

FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention;
FIG. 2 is a flow diagram of the code signing protocol described above with reference to FIG. 1;
FIG. 3 is a block diagram of a code signing system on a mobile device;
FIG. 3A is a block diagram of a code signing system on a plurality of mobile devices;
FIG. 4 is a flow diagram illustrating the operation of the code signing system described above with reference to FIG. 3 and FIG. 3A;
FIG. 5 is a flow diagram illustrating the management of the code signing authorities described with reference to FIG. 3A; and
FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented.

DETAILED DESCRIPTION

[0013] Referring now to the drawing figures, FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention. An application developer 12 creates a software application 14 (application Y) for a mobile device that requires access to one or more sensitive APIs on the mobile device. The software application Y 14 may, for example, be a Java application that operates on a Java virtual machine installed on the mobile device. An API enables the software application Y to interface with an application platform that may include, for example, resources such as the device hardware, operating system and core software and data models. In order to make function calls to or otherwise interact with such device resources, a software application Y must access one or more APIs. APIs can thereby effectively "bridge" a software application and associated device resources. In this description and the appended claims, references to API access should be interpreted to include access of an API in such a way as to allow a software application Y to interact with one or more corresponding device resources. Providing access to any API therefore allows a software application Y to interact with associated device resources, whereas denying access to an API prevents the software application Y from interacting with the associated resources. For example, a database API may communicate with a device file or data storage system, and access to the database API would provide for interaction between a software application Y and the file or data storage system. A user interface (UI) API would

communicate with controllers and/or control software for such device components as a screen, a keyboard, and any other device components that provide output to a user or accept input from a user. In a mobile device, a radio API may also be provided as an interface to wireless communication resources such as a transmitter and receiver. Similarly, a cryptographic API may be provided to interact with a crypto module which implements crypto algorithms on a device. These are merely illustrative examples of APIs that may be provided on a device. A device may include any of these example APIs, or different APIs instead of or in addition to those described above.
[0014] Preferably, any API may be classified as sensitive by a mobile device manufacturer, or possibly by an API author, a wireless network operator, a device owner or operator, or some other entity that may be affected by a virus or malicious code in a device software application. For instance, a mobile device manufacturer may classify as sensitive those APIs that interface with cryptographic routines, wireless communication functions, or proprietary data models such as address book or calendar entries. To protect against unauthorized access to these sensitive APIs, the application developer 12 is required to obtain one or more digital signatures from the mobile device manufacturer or other entity that classified any APIs as sensitive, or from a code signing authority 16 acting on behalf of the manufacturer or other entity with an interest in protecting access to sensitive device APIs, and append the signature(s) to the software application Y 14.
[0015] In one embodiment, a digital signature is obtained for each sensitive API or library that includes a sensitive API to which the software application requires access. In some cases, multiple signatures are desirable. This would allow a service provider, company or network operator to restrict some or all software applications loaded or updated onto a particular set of mobile devices. In this multiple-signature scenario, all APIs are restricted and locked until a "global" signature is verified for a software application. For example, a company may wish to prevent its employees from executing any software applications onto their devices without first obtaining permission from a corporate information technology (IT) or computer services department. All such corporate mobile devices may then be configured to require verification of at least a global signature before a software application can be executed. Access to sensitive device APIs and libraries, if any, could then be further restricted, dependent upon verification of respective corresponding digital signatures.
[0016] The binary executable representation of software application Y 14 may be independent of the particular type of mobile device or model of a mobile device. Software application Y 14 may for example be in a write-once-run-anywhere binary format such as is the case with Java software applications. However, it may be desirable to have a digital signature for each mobile device type or model, or alternatively for each mobile device

platform or manufacturer. Therefore, software application Y 14 may be submitted to several code signing authorities if software application Y 14 targets several mobile devices.

[0017] Software application Y 14 is sent from the application developer 12 to the code signing authority 16. In the embodiment shown in FIG. 1, the code signing authority 16 reviews the software application Y 14, although as described in further detail below, it is contemplated that the code signing authority 16 may also or instead consider the identity of the application developer 12 to determine whether or not the software application Y 14 should be signed. The code signing authority 16 is preferably one or more representatives from the mobile device manufacturer, the authors of any sensitive APIs, or possibly others that have knowledge of the operation of the sensitive APIs to which the software application needs access.

[0018] If the code signing authority 16 determines that software application Y 14 may access the sensitive API and therefore should be signed, then a signature (not shown) for the software application Y 14 is generated by the code signing authority 16 and appended to the software application Y 14. The signed software application Y 22, comprising the software application Y 14 and the digital signature, is then returned to the application developer 12. The digital signature is preferably a tag that is generated using a private signature key 18 maintained solely by the code signing authority 16. For example, according to one signature scheme, a hash of the software application Y 14 may be generated, using a hashing algorithm such as the Secure Hash Algorithm SHA1, and then used with the private signature key 18 to create the digital signature. In some signature schemes, the private signature key is used to encrypt a hash of information to be signed, such as software application Y 14, whereas in other schemes, the private key may be used in other ways to generate a signature from the information to be signed or a transformed version of the information.

[0019] The signed software application Y 22 may then be sent to a mobile device 28 or downloaded by the mobile device 28 over a wireless network 24. It should be understood, however, that a code signing protocol according to the present invention is not limited to software applications that are downloaded over a wireless network. For instance, in alternative embodiments, the signed software application Y 22 may be downloaded to a personal computer via a computer network and loaded to the mobile device through a serial link, or may be acquired from the application developer 12 in any other manner and loaded onto the mobile device. Once the signed software application Y 22 is loaded on the mobile device 28, each digital signature is preferably verified with a public signature key 20 before the software application Y 14 is granted access to a sensitive API library. Although the signed software application Y 22 is loaded onto a device, it should be appreciated that the software application that may eventually be executed on the device is the software application Y 14. As described above, the signed software application Y 22 includes the software application Y 14 and one or more appended digital signatures (not shown). When the signatures are verified, the software application Y 14 can be executed on the device and access any APIs for which corresponding signatures have been verified.

[0020] The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority 16, and is preferably installed on the mobile device along with the sensitive API. However, the public key 10 may instead be obtained from a public key repository (not shown), using the device 28 or possibly a personal computer system, and installed on the device 28 as needed. According to one embodiment of a signature scheme, the mobile device 28 calculates a hash of the software application Y 14 in the signed software application Y 22, using the same hashing algorithm as the code signing authority 16, and uses the digital signature and the public signature key 20 to recover the hash calculated by the signing authority 16. The resultant locally calculated hash and the hash recovered from the digital signature are then compared, and if the hashes are the same, the signature is verified. The software application Y 14 can then be executed on the device 28 and access any sensitive APIs for which the corresponding signature(s) have been verified. As described above, the invention is in no way limited to this particular illustrative example signature scheme. Other signature schemes, including further public key signature schemes, may also be used in conjunction with the code signing methods and systems described herein.

[0021] FIG. 2 is a flow diagram 30 of the code signing protocol described above with reference to FIG. 1. The protocol begins at step 32. At step 34, a software developer writes the software application Y for a mobile device that requires access to a sensitive API or library that exposes a sensitive API (API library A). As discussed above, some or all APIs on a mobile device may be classified as sensitive, thus requiring verification of a digital signature for access by any software application such as software application Y. In step 36, application Y is tested by the software developer, preferably using a device simulator in which the digital signature verification function has been disabled. In this manner, the software developer may debug the software application Y before the digital signature is acquired from the code signing authority. Once the software application Y has been written and debugged, it is forwarded to the code signing authority in step 38.

[0022] In steps 40 and 42, the code signing authority reviews the software application Y to determine whether or not it should be given access to the sensitive API, and either accepts or rejects the software application. The code signing authority may apply a number of criteria to determine whether or not to grant the software application access to the sensitive API including, for example, the size of the software application, the device resources

accessed by the API, the perceived utility of the software application, the interaction with other software applications, the inclusion of a virus or other destructive code, and whether or not the developer has a contractual obligation or other business arrangement with the mobile device manufacturer. Further details of managing code signing authorities and developers are described below in reference to FIG. 5.

[0023] If the code signing authority accepts the software application Y, then a digital signature, and preferably a signature identification, are appended to the software application Y in step 46. As described above, the digital signature may be generated by using a hash of the software application Y and a private signature key 18. The signature identification is described below with reference to FIGS. 3 and 4. Once the digital signature and signature identification are appended to the software application Y to generate a signed software application, the signed software application Y is returned to the software developer in step 48. The software developer may then license the signed software application Y to be loaded onto a mobile device (step 50). If the code signing authority rejects the software application Y, however, then a rejection notification is preferably sent to the software developer (step 44), and the software application Y will be unable to access any API(s) associated with the signature.

[0024] In an alternative embodiment, the software developer may provide the code signing authority with only a hash of the software application Y, or provide the software application Y in some type of abridged format. If the software application Y is a Java application, then the device independent binary *.class files may be used in the hashing operation, although device dependent files such as *.cod files used by the assignee of the present application may instead be used in hashing or other digital signature operations when software applications are intended for operation on particular devices or device types. By providing only a hash or abridged version of the software application Y, the software developer may have the software application Y signed without revealing proprietary code to the code signing authority. The hash of the software application Y, along with the private signature key 18, may then be used by the code signing authority to generate the digital signature. If an otherwise abridged version of the software application Y is sent to the code signing authority, then the abridged version may similarly be used to generate the digital signature, provided that the abridging scheme or algorithm, like a hashing algorithm, generates different outputs for different inputs. This ensures that every software application will have a different abridged version and thus a different signature that can only be verified when appended to the particular corresponding software application from which the abridged version was generated. Because this embodiment does not enable the code signing authority to thoroughly review the software application for viruses or other destructive code, however, a registration process

between the software developer and the code signing authority may also be required. For instance, the code signing authority may agree in advance to provide a trusted software developer access to a limited set of sensitive APIs.

[0025] In still another alternative embodiment, a software application Y may be submitted to more than one signing authority. Each signing authority may for example be responsible for signing software applications for particular sensitive APIs or APIs on a particular model of mobile device or set of mobile devices that supports the sensitive APIs required by a software application. A manufacturer, mobile communication network operator, service provider, or corporate client for example may thereby have signing authority over the use of sensitive APIs for their particular mobile device model(s), or the mobile devices operating on a particular network, subscribing to one or more particular services, or distributed to corporate employees. A signed software application may then include a software application and at least one appended digital signature appended from each of the signing authorities. Even though these signing authorities in this example would be generating a signature for the same software application, different signing and signature verification schemes may be associated with the different signing authorities.

[0026] FIG. 3 is a block diagram of a code signing system 60 on a mobile device 62. The system 60 includes a virtual machine 64, a plurality of software applications 66-70, a plurality of API libraries 72-78, and an application platform 80. The application platform 80 preferably includes all of the resources on the mobile device 62 that may be accessed by the software applications 66-70. For instance, the application platform may include device hardware 82, the mobile device's operating system 84, or core software and data models 86. Each API library 72-78 preferably includes a plurality of APIs that interface with a resource available in the application platform. For instance, one API library might include all of the APIs that interface with a calendar program and calendar entry data models. Another API library might include all of the APIs that interface with the transmission circuitry and functions of the mobile device 62. Yet another API library might include all of the APIs capable of interfacing with lower-level services performed by the mobile device's operating system 84. In addition, the plurality of API libraries 72-78 may include both libraries that expose a sensitive API 74 and 78, such as an interface to a cryptographic function, and libraries 72 and 76, that may be accessed without exposing sensitive APIs. Similarly, the plurality of software applications 66-70 may include both signed software applications 66 and 70 that require access to one or more sensitive APIs, and unsigned software applications such as 68. The virtual machine 64 is preferably an object oriented run-time environment such as Sun Micro System's J2ME(TM) (Java 2 Platform, Micro Edition), which manages the execution of all of the software applications 66-70 operating on the mobile de-

5

vice 62, and links the software applications 66-70 to the various API libraries 72-78.

[0027] Software application Y 70 is an example of a signed software application. Each signed software application preferably includes an actual software application such as software application Y comprising for example software code that can be executed on the application platform 80, one or more signature identifications 94 and one or more corresponding digital signatures 96. Preferably each digital signature 96 and associated signature identification 94 in a signed software application 66 or 70 corresponds to a sensitive API library 74 or 78 to which the software application X or software application Y requires access. The sensitive API library 74 or 78 may include one or more sensitive APIs. In an alternative embodiment, the signed software applications 66 and 70 may include a digital signature 96 for each sensitive API within an API library 74 or 78. The signature identifications 94 may be unique integers or some other means of relating a digital signature 96 to a specific API library 74 or 78, API, application platform 80, or model of mobile device 62.

[0028] API library A 78 is an example of an API library that exposes a sensitive API. Each API library 74 and 78 including a sensitive API should preferably include a description string 88, a public signature key 20, and a signature identifier 92. The signature identifier 92 preferably corresponds to a signature identification 94 in a signed software application 66 or 70, and enables the virtual machine 64 to quickly match a digital signature 96 with an API library 74 or 78. The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority, and is used to verify the authenticity of a digital signature 96. The description string 88 may for example be a textual message that is displayed on the mobile device when a signed software application 66 or 70 is loaded, or alternatively when a software application X or Y attempts to access a sensitive API.

[0029] Operationally, when a signed software application 68-70, respectively including a software application X, Z, or Y, that requires access to a sensitive API library 74 or 78 is loaded onto a mobile device, the virtual machine 64 searches the signed for an appended digital signature 96 associated with the API library 74 or 78. Preferably, the appropriate digital signature 96 is located by the virtual machine 64 by matching the signature identifier 92 in the API library 74 or 78 with a signature identification 94 on the signed software application. If the signed software application includes the appropriate digital signature 96, then the virtual machine 64 verifies its authenticity using the public signature key 20. Then, once the appropriate digital signature 96 has been located and verified, the description string 88 is preferably displayed on the mobile device before the software application X or Y is executed and accesses the sensitive API. For instance, the description string 88 may display a message stating that "Application Y is attempting to access

API Library A," and thereby provide the mobile device user with the final control to grant or deny access to the sensitive API.

[0030] FIG. 3A is a block diagram of a code signing system 61 on a plurality of mobile devices 62E, 62F and 62G. The system 61 includes a plurality of mobile devices each of which only three are illustrated, mobile devices 62E, 62F and 62G. Also shown is a signed software application 70, including a software application Y to which two digital signatures 96E and 96F with corresponding signature identifications 94E and 94F have been appended. In the example system 61, each pair composed of a digital signature and identification, 94E/96E and 94F/96F, corresponds to a model of mobile device 62, API library 78, or associated platform 80. If signature identifications 94E and 94F correspond to different models of mobile device 62, then when a signed software application 70 which includes a software application Y that requires access to a sensitive API library 78 is loaded onto mobile device 62E, the virtual machine 64 searches the signed software application 70 for a digital signature 96E associated with the API library 78 by matching identifier 94E with signature identifier 92. Similarly, when a signed software application 70 including a software application Y that requires access to a sensitive API library 78 is loaded onto a mobile device 62F, the virtual machine 64 in device 62F searches the signed software application 70 for a digital signature 96F associated with the API library 78. However, when a software application Y in a signed software application 70 that requires access to a sensitive API library 78 is loaded onto a mobile device model for which the application developer has not obtained a digital signature, device 62G in the example of FIG. 3A, the virtual machine 64 in the device 64G does not find a digital signature appended to the software application Y and consequently, access to the API library 78 is denied on device 62G. It should be appreciated from the foregoing description that a software application such as software application Y may have multiple device-specific, library-specific, or API-specific signatures or some combination of such signatures appended thereto. Similarly, different signature verification requirements may be configured for the different devices. For example, device 62E may require verification of both a global signature, as well as additional signatures for any sensitive APIs to which a software application, requires access in order for the software application to be executed, whereas device 62F may require verification of only a global signature and device 62G may require verification of signatures only for its sensitive APIs. It should also be apparent that a communication system may include devices (not shown) on which a software application Y received as part of a signed software application such as 70 may execute without any signature verification. Although a signed software application has one or more signatures appended thereto, the software application Y might possibly be executed on some devices without first having any of its signature(s) verified. Signing of a software ap-

6

plication preferably does not interfere with its execution on devices in which digital signature verification is not implemented.

[0031] FIG. 4 is a flow diagram 100 illustrating the operation of the code signing system described above with reference to FIGS. 3 and 3A. In step 102, a software application is loaded onto a mobile device. Once the software application is loaded, the device, preferably using a virtual machine, determines whether or not the software application requires access to any API libraries that expose a sensitive API (step 104). If not, then the software application is linked with all of its required API libraries and executed (step 118). If the software application does require access to a sensitive API, however, then the virtual machine verifies that the software application includes a valid digital signature associated any sensitive APIs to which access is required, in steps 106-116.

[0032] In step 106, the virtual machine retrieves the public signature key 20 and signature identifier 92 from the sensitive API library. The signature identifier 92 is then used by the virtual machine in step 108 to determine whether or not the software application has an appended digital signature 96 with a corresponding signature identification 94. If not, then the software application has not been approved for access to the sensitive API by a code signing authority, and the software application is preferably prevented from being executed in step 116. In alternative embodiments, a software application without a proper digital signature 96 may be purged from the mobile device, or may be denied access to the API library exposing the sensitive API but executed to the extent possible without access to the API library. It is also contemplated that a user may be prompted for an input when signature verification fails, thereby providing for user control of such subsequent operations as purging of the software application from the device.

[0033] If a digital signature 96 corresponding to the sensitive API library is appended to the software application and located by the virtual machine, then the virtual machine uses the public key 20 to verify the authenticity of the digital signature 96 in step 110. This step may be performed, for example, by using the signature verification scheme described above or other alternative signature schemes. If the digital signature 96 is not authentic, then the software application is preferably either not executed, purged, or restricted from accessing the sensitive API as described above with reference to step 116. If the digital signature is authentic, however, then the description string 88 is preferably displayed in step 112, warning the mobile device user that the software application requires access to a sensitive API, and possibly prompting the user for authorization to execute or load the software application (step 114). When more than one signature is to be verified for a software application, then the steps 104-110 are preferably repeated for each signature before the user is prompted in step 112. If the mobile device user in step 114 authorizes the software application, then it may be executed and linked to the sensitive API library

in step 118.

[0034] FIG. 5 is a flow diagram 200 illustrating the management of the code signing authorities described with reference to FIG. 3A. At step 210, an application developer has developed a new software application which is intended to be executable one or more target device models or types. The target devices may include sets of devices from different manufacturers, sets of device models or types from the same manufacturer, or generally any sets of devices having particular signature and verification requirements. The term "target device" refers to any such set of devices having a common signature requirement. For example, a set of devices requiring verification of a device-specific global signature for execution of all software applications may comprise a target device, and devices that require both a global signature and further signatures for sensitive APIs may be part of more than one target device set. The software application may be written in a device independent manner by using at least one known API, supported on at least one target device with an API library. Preferably, the developed software application is intended to be executable on several target devices, each of which has its own at least one API library.

[0035] At step 220, a code signing authority for one target device receives a target-signing request from the developer. The target signing request includes the software application or a hash of the software application, a developer identifier, as well as at least one target device identifier which identifies the target device for which a signature is being requested. At step 230, the signing authority consults a developer database 235 or other records to determine whether or not to trust developer 220. This determination can be made according to several criteria discussed above, such as whether or not the developer has a contractual obligation or has entered into some other type of business arrangement with a device manufacturer, network operator, service provider, or device manufacturer. If the developer is trusted, then the method proceeds at step 240. However, if the developer is not trusted, then the software application is rejected (250) and not signed by the signing authority. Assuming the developer was trusted, at step 240 the signing authority determines if it has the target private key corresponding to the submitted target identifier by consulting a private key store such as a target private key database 245. If the target private key is found, then a digital signature for the software application is generated at step 260 and the digital signature or a signed software application including the digital signature appended to the software application is returned to the developer at step 280. However, if the target private key is not found at step 240, then the software application is rejected at step 270 and no digital signature is generated for the software application.

[0036] Advantageously, if target signing authorities follow compatible embodiments of the method outlined in FIG. 5, a network of target signing authorities may be established in order to expediently manage code signing

authorities and a developer community code signing process providing signed software applications for multiple targets with low likelihood of destructive code.

[0037] Should any destructive or otherwise problematic code be found in a software application or suspected because of behavior exhibited when a software application is executed on a device, then the registration or privileges of the corresponding application developer with any or all signing authorities may also be suspended or revoked, since the digital signature provides an audit trail through which the developer of a problematic software application may be identified. In such an event, devices may be informed of the revocation by being configured to periodically download signature revocation lists, for example. If software applications for which the corresponding digital signatures have been revoked are running on a device, the device may then halt execution of any such software application and possibly purge the software application from its local storage. If preferred, devices may also be configured to re-execute digital signature verifications, for instance periodically or when a new revocation list is downloaded.

[0038] Although a digital signature generated by a signing authority is dependent upon authentication of the application developer and confirmation that the application developer has been properly registered, the digital signature is preferably generated from a hash or otherwise transformed version of the software application and is therefore application-specific. This contrasts with known code signing schemes, in which API access is granted to any software applications arriving from trusted application developers or authors. In the code signing systems and methods described herein, API access is granted on an application-by-application basis and thus can be more strictly controlled or regulated.

[0039] FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented. The mobile communication device 610 is preferably a two-way communication device having at least voice and data communication capabilities. The device preferably has the capability to communicate with other computer systems on the Internet. Depending on the functionality provided by the device, the device may be referred to as a data messaging device, a two-way pager, a cellular telephone with data messaging capabilities, a wireless Internet appliance or a data communication device (with or without telephony capabilities).

[0040] Where the device 610 is enabled for two-way communications, the device will incorporate a communication subsystem 611, including a receiver 612, a transmitter 614, and associated components such as one or more, preferably embedded or internal, antenna elements 616 and 618, local oscillators (LOs) 613, and a processing module such as a digital signal processor (DSP) 620. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem 611 will be dependent upon the com-

munication network in which the device is intended to operate. For example, a device 610 destined for a North American market may include a communication subsystem 611 designed to operate within the Mobitex(TM) mobile communication system or DataTAC(TM) mobile communication system, whereas a device 610 intended for use in Europe may incorporate a General Packet Radio Service (GPRS) communication subsystem 611.

[0041] Network access requirements will also vary depending upon the type of network 919. For example, in the Mobitex and DataTAC networks, mobile devices such as 610 are registered on the network using a unique identification number associated with each device. In GPRS networks however, network access is associated with a subscriber or user of a device 610. A GPRS device therefore requires a subscriber identity module (not shown), commonly referred to as a SIM card, in order to operate on a GPRS network. Without a SIM card, a GPRS device will not be fully functional. Local or non-network communication functions (if any) may be operable, but the device 610 will be unable to carry out any functions involving communications over network 619, other than any legally required operations such as "911" emergency calling.

[0042] When required network registration or activation procedures have been completed, a device 610 may send and receive communication signals over the network 619. Signals received by the antenna 616 through a communication network 619 are input to the receiver 612, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in FIG. 6, analog to digital conversion. Analog to digital conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP 620. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by the DSP 620 and input to the transmitter 614 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission over the communication network 619 via the antenna 618.

[0043] The DSP 620 not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in the receiver 612 and transmitter 614 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 620.

[0044] The device 610 preferably includes a microprocessor 638 which controls the overall operation of the device. Communication functions, including at least data and voice communications, are performed through the communication subsystem 611. The microprocessor 638 also interacts with further device subsystems or resources such as the display 622, flash memory 624, random access memory (RAM) 626, auxiliary input/output (I/O) subsystems 628, serial port 630, keyboard 632, speaker 634, microphone 636, a short-range communications subsystem 640 and any other device subsystems gen-

erally designated as 642. APIs, including sensitive APIs requiring verification of one or more corresponding digital signatures before access is granted, may be provided on the device 610 to interface between software applications and any of the resources shown in FIG. 6.

[0045] Some of the subsystems shown in FIG. 6 perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard 632 and display 622 for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0046] Operating system software used by the microprocessor 638, and possibly APIs to be accessed by software applications, is preferably stored in a persistent store such as flash memory 624, which may instead be a read only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device software applications, or parts thereof, may be temporarily loaded into a volatile store such as RAM 626. It is contemplated that received and transmitted communication signals may also be stored to RAM 626.

[0047] The microprocessor 638, in addition to its operating system functions, preferably enables execution of software applications on the device. A predetermined set of applications which control basic device operations, including at least data and voice communication applications for example, will normally be installed on the device 610 during manufacture. A preferred application that may be loaded onto the device may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the device user such as, but not limited to e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the device to facilitate storage of PIM data items on the device. Such PIM application would preferably have the ability to send and receive data items, via the wireless network. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network, with the device user's corresponding data items stored or associated with a host computer system thereby creating a mirrored host computer on the mobile device with respect to the data items at least. This would be especially advantageous in the case where the host computer system is the mobile device user's office computer system. Further applications, including signed software applications as described above, may also be loaded onto the device 610 through the network 619, an auxiliary I/O subsystem 62S, serial port 630, short-range communications subsystem 640 or any other suitable subsystem 642. The device microprocessor 638 may then verify any digital signatures, possibly including both "global" device signatures and API-specific signatures, appended to such a software application before the soft-

ware application can be executed by the microprocessor 638 and/or access any associated sensitive APIs. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the device 610, through a crypto API and a crypto module which implements crypto algorithms on the device (not shown).

[0048] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem 611 and input to the microprocessor 638, which will preferably further process the received signal for output to the display 622, or alternatively to an auxiliary I/O device 628. A user of device 610 may also compose data items such as email messages for example, using the keyboard 632, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display 622 and possibly an auxiliary I/O device 628. Such composed items may then be transmitted over a communication network through the communication subsystem 611.

[0049] For voice communications, overall operation of the device 610 is substantially similar, except that received signals would preferably be output to a speaker 634 and signals for transmission would be generated by a microphone 636. Alternative voice or audio I/O subsystems such as a voice message recording subsystem may also be implemented on the device 610. Although voice or audio signal output is preferably accomplished primarily through the speaker 634, the display 622 may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information for example.

[0050] The serial port 630 in FIG. 6 would normally be implemented in a personal digital assistant (PDA)-type communication device for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port 630 would enable a user to set preferences through an external device or software application and would extend the capabilities of the device by providing for information or software downloads to the device 610 other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the device through a direct and thus reliable and trusted connection to thereby enable secure device communication.

[0051] A short-range communications subsystem 640 is a further optional component which may provide for communication between the device 624 and different systems or devices, which need not necessarily be similar devices. For example, the subsystem 640 may include an infrared device and associated circuits and components or a Bluetooth(TM) communication module to provide for communication with similarly-enabled sys-

tems and devices.

[0052] The embodiments described herein are examples of structures, systems or methods having elements corresponding to the elements of the invention recited in the claims. This written description may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements of the invention recited in the claims. The intended scope of the invention thus includes other structures, systems or methods that do not differ from the literal language of the claims, and further includes other structures, systems or methods with insubstantial differences from the literal language of the claims.

[0053] For example, when a software application is rejected at step 250 in the method shown in FIG. 5, the signing authority may request that the developer sign a contract or enter into a business relationship with a device manufacturer or other entity on whose behalf the signing authority acts. Similarly, if a software application is rejected at step 270, authority to sign the software application may be delegated to a different signing authority. The signing of a software application following delegation of signing of the software application to the different authority can proceed substantially as shown in FIG. 5, wherein the target signing authority that received the original request from the trusted developer at step 220 requests that the software application be signed by the different signing authority on behalf of the trusted developer from the target signing authority. Once a trust relationship has been established between code signing authorities, target private code signing keys could be shared between code signing authorities to improve performance of the method at step 240, or a device may be configured to validate digital signatures from either of the trusted signing authorities.

[0054] In addition, although described primarily in the context of software applications, code signing systems and methods according to the present invention may also be applied to other device-related components, including but in no way limited to, commands and associated command arguments, and libraries configured to interface with device resources. Such commands and libraries may be sent to mobile devices by device manufacturers, device owners, network operators, service providers, software application developers and the like. It would be desirable to control the execution of any command that may affect device operation, such as a command to change a device identification code or wireless communication network address for example, by requiring verification of one or more digital signatures before a command can be executed on a device, in accordance with the code signing systems and methods described and claimed herein.

[0055] As has been described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; an application programming interface (API) configured to link the software application with the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0056] The virtual machine may deny the software application access to the API if the digital signature is not authentic. The virtual machine may purge the software application if the digital signature is not authentic. The code signing system may be installed on a mobile device. The digital signature may be generated by a code signing authority.

[0057] The code signing system may further comprise a plurality of API libraries each including a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

[0058] One or more of the plurality of API libraries may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive API libraries by the software application. The software application may include a unique digital signature for each sensitive API library. The software application may include a signature identification for each unique digital signature; each sensitive API library may include a signature identifier; and the virtual machine may compare the signature identification and the signature identifier to match the unique digital signatures with sensitive API libraries.

[0059] The digital signature may be generated using a private signature key, and the virtual machine may use a public signature key to verify the authenticity of the digital signature. The digital signature may be generated by applying the private signature key to a hash of the software application; and the virtual machine may verify the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

[0060] The API may further comprise a description string that is displayed by the mobile device when the software application attempts to access the API. The application platform may comprise an operating system. The application platform may comprise one or more core functions of a mobile device. The application platform may comprise hardware on a mobile device. The hardware may comprise a subscriber identity module (SIM) card. The software application may be a Java application for a mobile device. The API may interface with a cryptographic routine on the application platform. The API may interface with a proprietary data model on the application platform. The virtual machine may be a Java virtual machine installed on a mobile device.

[0061] As also described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; a plurality of application programming interfaces (APIs), each configured to link the software application with a resource on the application platform; and a virtual machine that verifies the authenticity of the digital signature

in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0062] The plurality of APIs may be included in an API library. One or more of the plurality of APIs may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive APIs. For operation in conjunction with a plurality of software applications, one or more of the plurality of software applications may have a digital signature, and the virtual machine may verify the authenticity of the digital signature of each of the one or more of the plurality of software applications in order to control access to the sensitive APIs by each of the plurality of software applications. The resource on the application platform may comprise a wireless communication system. The resource on the application platform may comprise a cryptographic module which implements cryptographic algorithms. The resource on the application platform may comprise a data store. The resource on the application platform may comprise a user interface (UI).

[0063] As has also been described, a method of controlling access to sensitive application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to a sensitive application programming interface (API); determining whether or not the software application includes a digital signature associated with the sensitive API; and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0064] The method my comprise the additional step of: if the software application does not include a digital signature associated with the sensitive API, then purging the software application from the mobile device. The digital signature may be generated by a code signing authority. The method may comprise the additional steps of: if the software application includes a digital signature associated with the sensitive API, then verifying the authenticity of the digital signature; and if the digital signature is not authentic, then denying the software application access to the sensitive API. The method may further comprise the additional step of: if the digital signature is not authentic, then purging the software application from the mobile device. The digital signature may be generated by applying a private signature key to a hash of the software application, and the step of verifying the authenticity of the digital signature may be performed by a method comprising the steps of: storing a public signature key that corresponds to the private signature key on the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash. The digital signature may be generated by calculating a hash of the software application and applying the

private signature key. The method may comprise the additional step of: displaying a description string that notifies a user of the mobile device that the software application requires access to the sensitive API. The method may further comprise the additional step of: receiving a command from the user granting or denying the software application access to the sensitive API.

[0065] Further has been described a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer, comprising the steps of: receiving the software application from the software developer; reviewing the software application to determine if it may access the API; if the software application may access the API, then appending a digital signature to the software application; verifying the authenticity of a digital signature appended to a software application; and providing access to the API to software applications for which the appended digital signature is authentic.

[0066] The step of reviewing the software application may be performed by a code signing authority. The step of appending the digital signature to the software application may be performed by a method comprising the steps of: calculating a hash of the software application; and applying a signature key to the hash of the software application to generate the digital signature. The hash of the software application may be calculated using the Secure Hash Algorithm (SHA1). The step of verifying the authenticity of a digital signature may comprise the steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; and determining if the digital signature is authentic by comparing the calculated hash with the recovered hash. The method may further comprise the step of, if the digital signature is not authentic, then denying the software application access to the API. The signature key may be a private signature key and the corresponding signature key is a public signature key.

[0067] Also has been described, a method of controlling access to a sensitive application programming interface (API) on a mobile device, comprising the steps of: registering one or more software developers that are trusted to design software applications which access the sensitive API; receiving a hash of a software application; determining if the software application was designed by one of the registered software developers; and if the software application was designed by one of the registered software developers, then generating a digital signature using the hash of the software application, wherein the digital signature may be appended to the software application; and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0068] The step of generating the digital signature may be performed by a code signing authority. The step of generating the digital signature may be performed by ap-

plying a signature key to the hash of the software application. The mobile device may verify the authenticity of the digital signature by performing the additional steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; determining if the digital signature is authentic by comparing the calculated hash with the recovered hash; and if the digital signature is not authentic, then denying the software application access to the sensitive API.

[0069] As has been described, a method of restricting access to application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to one or more application programming interface (API); determining whether or not the software application includes an authentic digital signature associated with the mobile device; and if the software application does not include an authentic digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

[0070] The method may comprise the additional step of: if the software application does not include an authentic digital signature associated with the mobile device, then purging the software application from the mobile device. The software application may include a plurality of digital signatures. The plurality of digital signatures may include digital signatures respectively associated with different types of mobile devices.

[0071] Each of the plurality of digital signatures may be generated by a respective corresponding code signing authority. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the additional steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature. The one or more APIs may include one or more APIs classified as sensitive, and the method may further comprise the steps of, for each sensitive API: determining whether or not the software application includes an authentic digital signature associated with the sensitive API; and if the software application does not include an authentic digital signature associated with the sensitive API, then denying the software application access to the sensitive API. Each of the plurality of digital signatures may be generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature, wherein the step of verifying the

authenticity of the digital signature is performed by a method comprising the steps of: storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the signature associated with the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash.

Claims

1. A method of controlling access to a sensitive application programming interface (API) having a signature identifier on a mobile device (62), comprising the steps of:

   registering one or more software developers that are trusted to develop software application (66) which access the sensitive API;
   receiving a hash of a software application;
   determining whether the hash was sent by a registered software developer; and
   generating a digital signature (96) using the hash of the software application (66) and a signature identification (94) corresponding to the signature identifier (92) where the hash was sent by the registered software developer; wherein the digital signature (96) and the signature identification (94) are appended to the software application (66); and
   the mobile device (62) verifies the authenticity of the digital signature (96) in order to control access to the sensitive API by the software application (66) where the signature identification (94) corresponds with the signature identifier (92).

2. The method of claim 1, wherein the step of generating the digital signature (96) is performed by a code signing authority.

3. The method of claim 1 or 2, wherein the step of generating the digital signature (96) is performed by applying a signature key to the hash of the software application (66).

4. The method of claim 3, wherein the mobile device (62) verifies the authenticity of the digital signature (96) by performing the additional steps of:

   providing a corresponding signature key on the mobile device (62);
   calculating the hash of the software application (66) on the mobile device (62) to obtain a calulated hash;

applying the corresponding signature key to the digital signature to obtain a recovered hash; determining whether the digital signature (96) is authentic by comparing the calculated hash with the recovered hash; and

denying the software application (66) access to where the digital signature (96) is not authenticated.

5. The method of any preceding claim, wherein a virtual machine (64) that verifies the authenticity of the digital signature (96) is provided.

6. The method of claim 5, wherein the virtual machine (64) denies the software application (66) access to the API if the digital signature (96) is not authenticated.

7. The method of claim 5, wherein the virtual machine (64) purges the software application (66) if the digital signature (96) is not authenticated.

8. The method of claim 5, wherein the digital signature (96) is generated using a private signature key, and the virtual machine (64) uses a public signature key to verify the authenticity of the digital signature (96).

9. The method of claim 8, wherein:

    the digital signature (96) is generated by applying the private signature key to a hash of the software application (66); and
    the virtual machine (64) verifies the authenticity of the digital signature (96) by generating a hash of the software application (66) to obtain a generated hash, applying the public signature key to the digital signature (96) to obtain a recovered hash, and comparing the generated hash with the recovered hash.

Patentansprüche

1. Verfahren zum Steuern eines Zugangs zu einer sensitiven Anwendungsprogrammierungsschnittstelle (API - application programming interface) mit einem Signaturidentifizierer auf einer mobilen Vorrichtung (62), das die Schritte aufweist:

    Registrieren eines oder mehrerer Softwareentwickler(s), der/die vertrauenswürdig ist/sind, um Softwareanwendungen (66) zu entwickeln, die auf die sensitive API zugreifen;
    Empfangen eines Hashs einer Softwareanwendung;
    Bestimmen, ob der Hash durch einen registrierten Softwareentwickler gesendet wurde; und
    Erzeugen einer digitalen Signatur (96) unter

Verwendung des Hashs der Softwareanwendung (66) und einer Signaturidentifikation (94), die dem Signaturidentifizierer (92) entspricht, wenn der Hash durch den registrierten Softwareentwickler gesendet wurde; wobei

die digitale Signatur (96) und die Signaturidentifikation (94) an die Softwareanwendung (66) angefügt werden; und
die mobile Vorrichtung (62) die Authentizität der digitalen Signatur (96) verifiziert, um einen Zugang zu der sensitiven API durch die Softwareanwendung (66) zu steuern, wenn die Signaturidentifikation (94) dem Signaturidentifizierer (92) entspricht.

2. Verfahren gemäß Anspruch 1, wobei der Schritt des Erzeugens der digitalen Signatur (96) durch eine Codesignierautorität durchgeführt wird.

3. Verfahren gemäß Anspruch 1 oder 2, wobei der Schritt des Erzeugens der digitalen Signatur (96) durchgeführt wird durch Anwenden eines Signaturschlüssels auf den Hash der Softwareanwendung (66).

4. Verfahren gemäß Anspruch 3, wobei die mobile Vorrichtung (62) die Authentizität der digitalen Signatur (96) verifiziert durch Durchführen der zusätzlichen Schritte:

    Vorsehen eines entsprechenden Signaturschlüssels auf der mobilen Vorrichtung (62);
    Berechnen des Hashs der Softwareanwendung (66) auf der mobilen Vorrichtung (62), um einen berechneten Hash zu erlangen;
    Anwenden des entsprechenden Signaturschlüssels auf die digitale Signatur, um einen wiedergewonnenen Hash zu erlangen;
    Bestimmen, ob die digitale Signatur (96) authentisch ist durch Vergleichen des berechneten Hashs mit dem wiedergewonnenen Hash; und
    Verweigern eines Zugangs zu der sensitiven API für die Softwareanwendung (66), wenn die digitale Signatur (96) nicht authentisiert ist.

5. Verfahren gemäß einem beliebigen vorangegangenen Anspruch, wobei eine virtuelle Maschine (64) vorgesehen ist, welche die Authentizität der digitalen Signatur (96) verifiziert.

6. Verfahren gemäß Anspruch 5, wobei die virtuelle Maschine (64) der Softwareanwendung (66) einen Zugang zur der API verweigert, wenn die digitale Signatur (96) nicht authentisiert ist.

7. Verfahren gemäß Anspruch 5, wobei die virtuelle Maschine (64) die Softwareanwendung (66) entfernt, wenn die digitale Signatur (96) nicht authenti-

siert ist.

8. Verfahren gemäß Anspruch 5, wobei die digitale Signatur (96) unter Verwendung eines privaten Signaturschlüssels erzeugt wird und die virtuelle Maschine (64) einen öffentlichen Signaturschlüssel verwendet, um die Authentizität der digitalen Signatur (96) zu verifizieren.

9. Verfahren gemäß Anspruch 8, wobei:

die digitale Signatur (96) erzeugt wird durch Anwenden des privaten Signaturschlüssels auf einen Hash der Softwareanwendung (66); und die virtuelle Maschine (64) die Authentizität der digitalen Signatur (96) verifiziert durch Erzeugen eines Hashs der Softwareanwendung (66), um einen erzeugten Hash zu erlangen, Anwenden des öffentlichen Signaturschlüssels auf die digitale Signatur (96), um einen wiedergewonnenen Hash zu erlangen, und Vergleichen des erzeugten Hashs mit dem wiedergewonnenen Hash.

Revendications

1. Procédé pour commander l'accès à une interface de programmation applicative (API) sensible ayant un identifiant de signature sur un dispositif mobile (62), comprenant les étapes qui consistent à :

enregistrer un ou plusieurs développeurs informatiques de confiance pour le développement d'applications logicielles (66) qui accèdent à l'API sensible ; recevoir une empreinte numérique d'une application logicielle ; déterminer si l'empreinte numérique a été envoyée par un développeur informatique enregistré ; et générer une signature numérique (96) en utilisant l'empreinte numérique de l'application logicielle (66) et une identification de signature (94) correspondant à l'identifiant de signature (92) où l'empreinte numérique a été envoyée par le développeur informatique enregistré ; où la signature numérique (96) et l'identification de signature (94) sont annexées à l'application logicielle (66) ; et le dispositif mobile (62) vérifie l'authenticité de la signature numérique (96) afin de commander l'accès à l'API sensible par l'application logicielle (66) où l'identification de signature (94) correspond à l'identifiant de signature (92).

2. Procédé de la revendication 1, dans lequel l'étape qui consiste à générer la signature numérique (96)

est exécutée par une autorité de signature de code.

3. Procédé de la revendication 1 ou 2, dans lequel l'étape qui consiste à générer la signature numérique (96) est exécutée en appliquant une clé de signature à l'empreinte numérique de l'application logicielle (66).

4. Procédé de la revendication 3, dans lequel le dispositif mobile (62) vérifie l'authenticité de la signature numérique (96) en exécutant les étapes supplémentaires qui consistent à :

fournir une clé de signature correspondante sur le dispositif mobile (62) ; calculer l'empreinte numérique de l'application logicielle (66) sur le dispositif mobile (62) pour obtenir une empreinte numérique calculée ; appliquer la clé de signature correspondante à la signature numérique pour obtenir une empreinte numérique récupérée ; déterminer si la signature numérique (96) est authentique en comparant l'empreinte numérique calculée à l'empreinte numérique récupérée ; et refuser à l'application logicielle (66) l'accès à l'API sensible lorsque la signature numérique (96) n'est pas authentifiée.

5. Procédé d'une des revendications précédentes, dans lequel une machine virtuelle (64) qui vérifie l'authenticité de la signature numérique (96) est prévue.

6. Procédé de la revendication 5, dans lequel la machine virtuelle (64) refuse à l'application logicielle (66) l'accès à l'API si la signature numérique (96) n'est pas authentifiée.

7. Procédé de la revendication 5, dans lequel la machine virtuelle (64) élimine l'application logicielle (66) si la signature numérique (96) n'est pas authentifiée.

8. Procédé de la revendication 5, dans lequel la signature numérique (96) est générée en utilisant une clé de signature privée, et la machine virtuelle (64) utilise une clé de signature publique pour vérifier l'authenticité de la signature numérique (96).

9. Procédé de la revendication 8, dans lequel :

la signature numérique (96) est générée en appliquant la clé de signature privée à une empreinte numérique de l'application logicielle (66) ; et la machine virtuelle (64) vérifie l'authenticité de la signature numérique (96) en générant une empreinte numérique de l'application logicielle

(66) afin d'obtenir une empreinte numérique gé-
nérée, en appliquant la clé de signature publique
à la signature numérique (96) afin d'obtenir une
empreinte numérique récupérée, et en compa-
rant l'empreinte numérique générée à l'emprein-
te numérique récupérée.

5

10

15

20

25

30

35

40

45

50

55

15
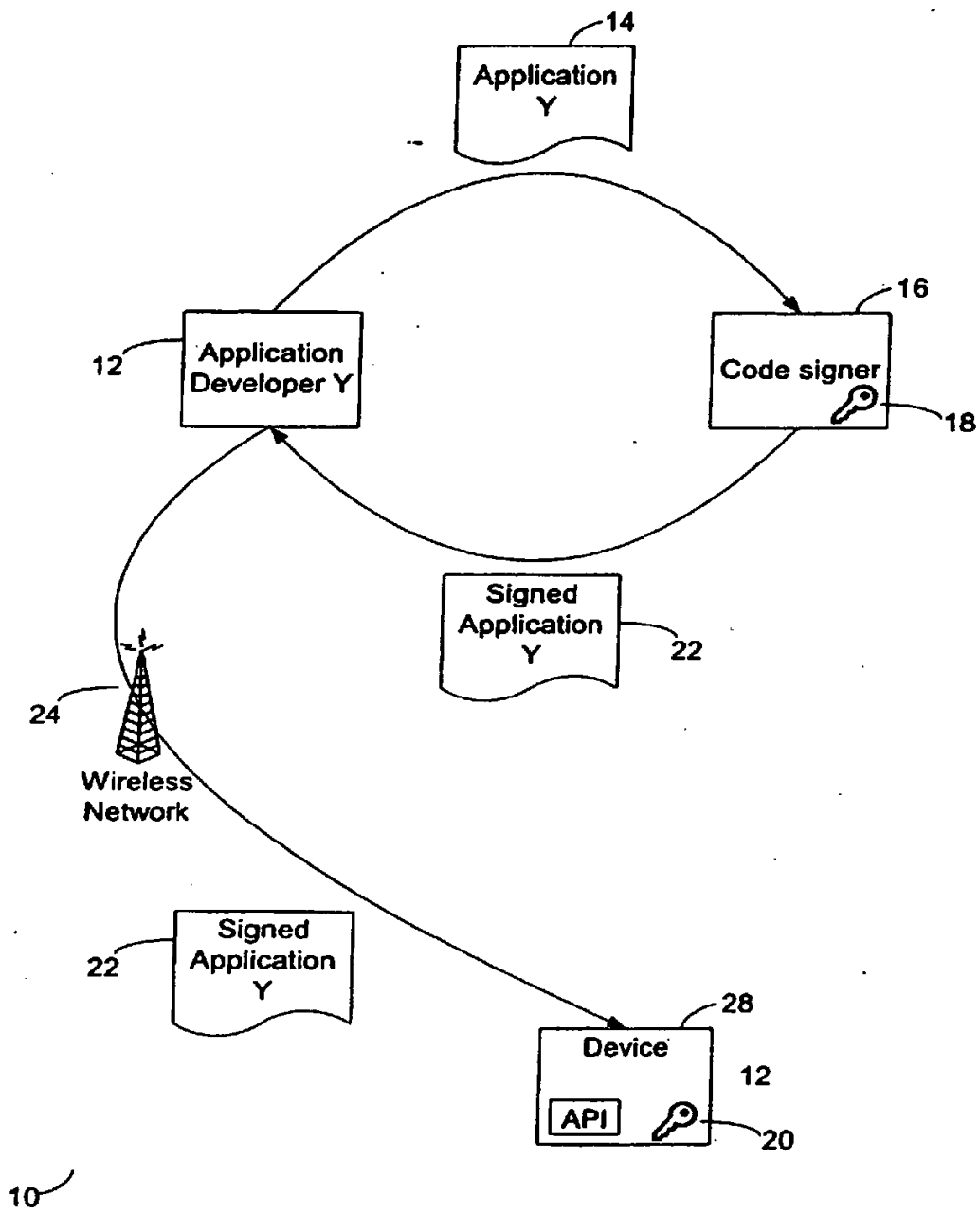
Figure 1

Figure 2
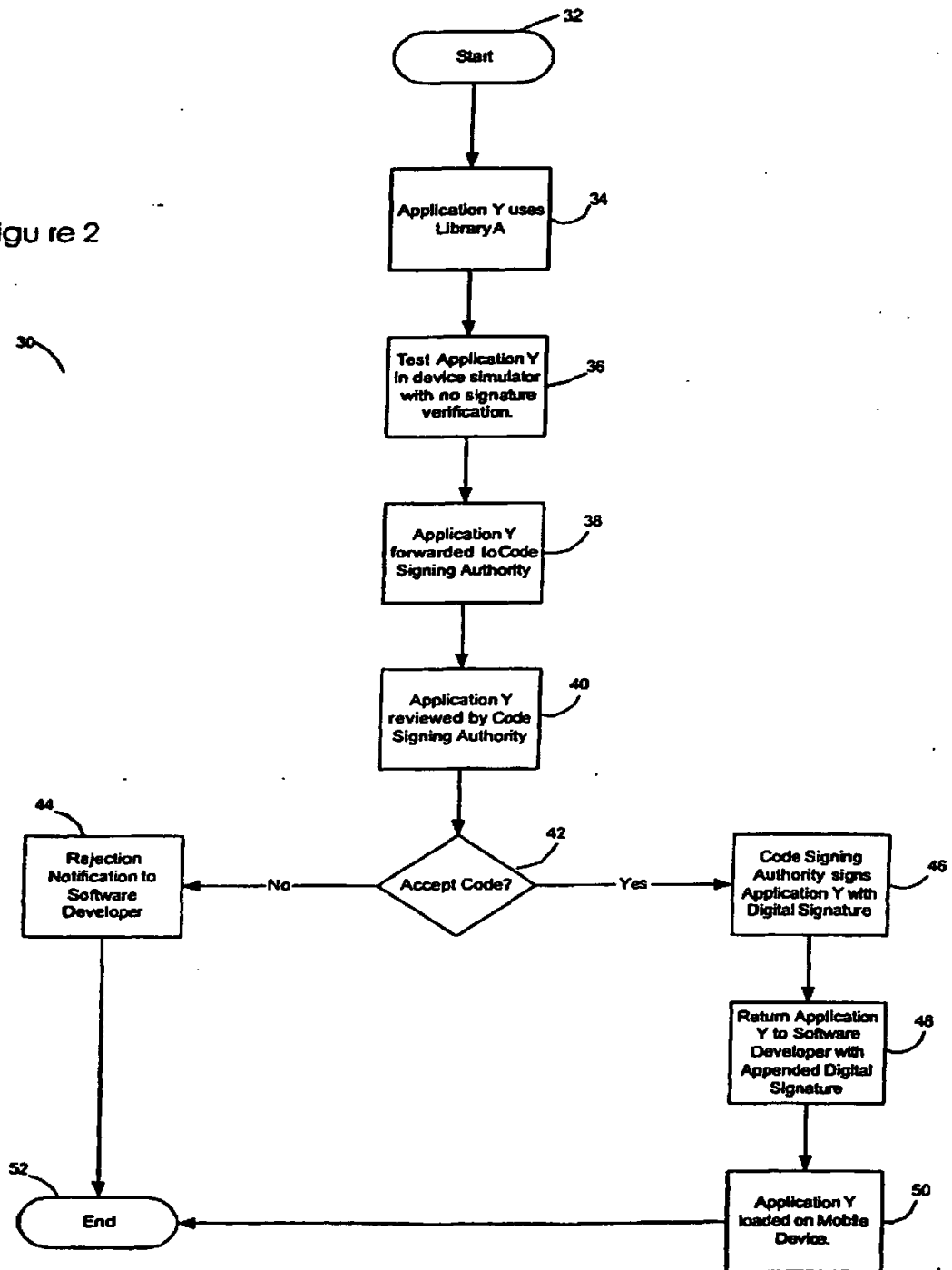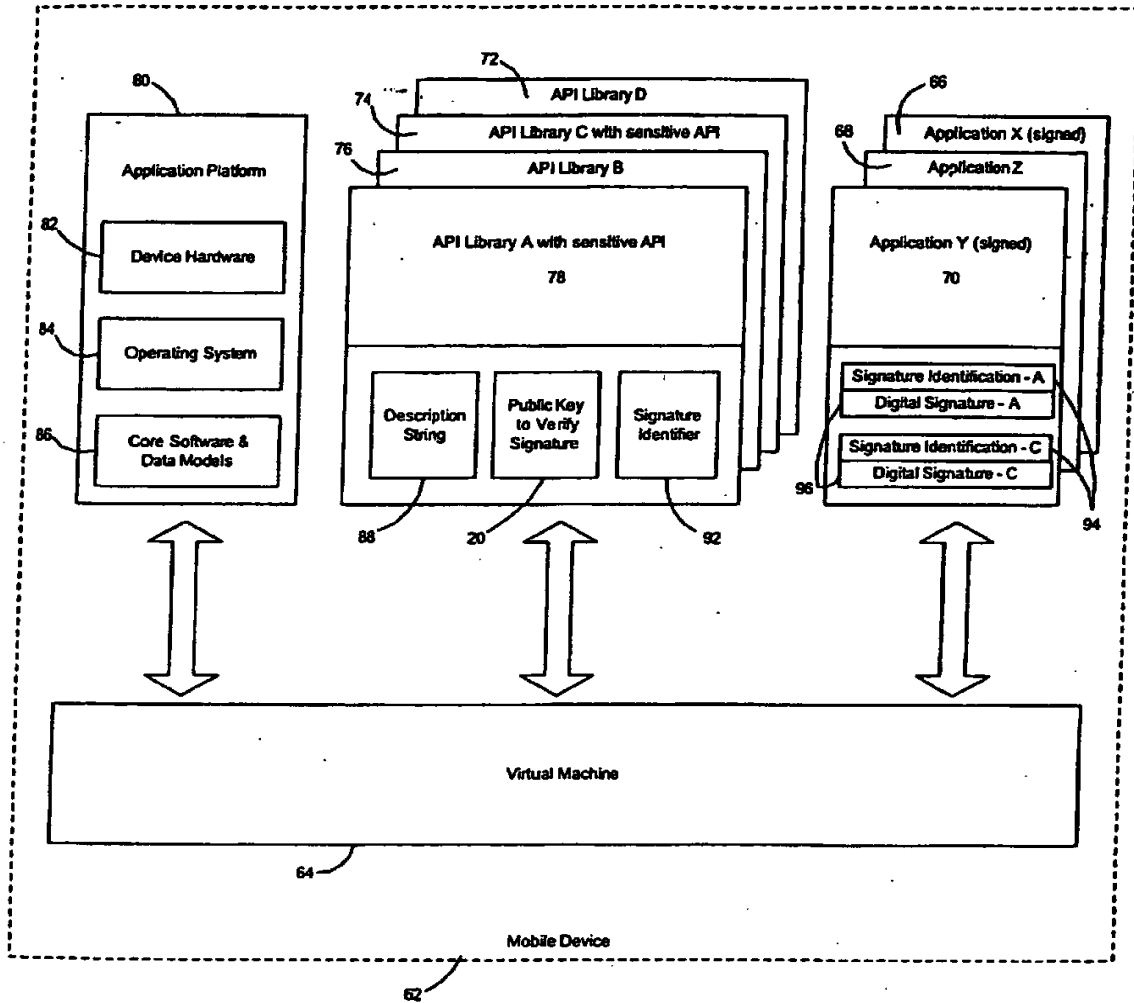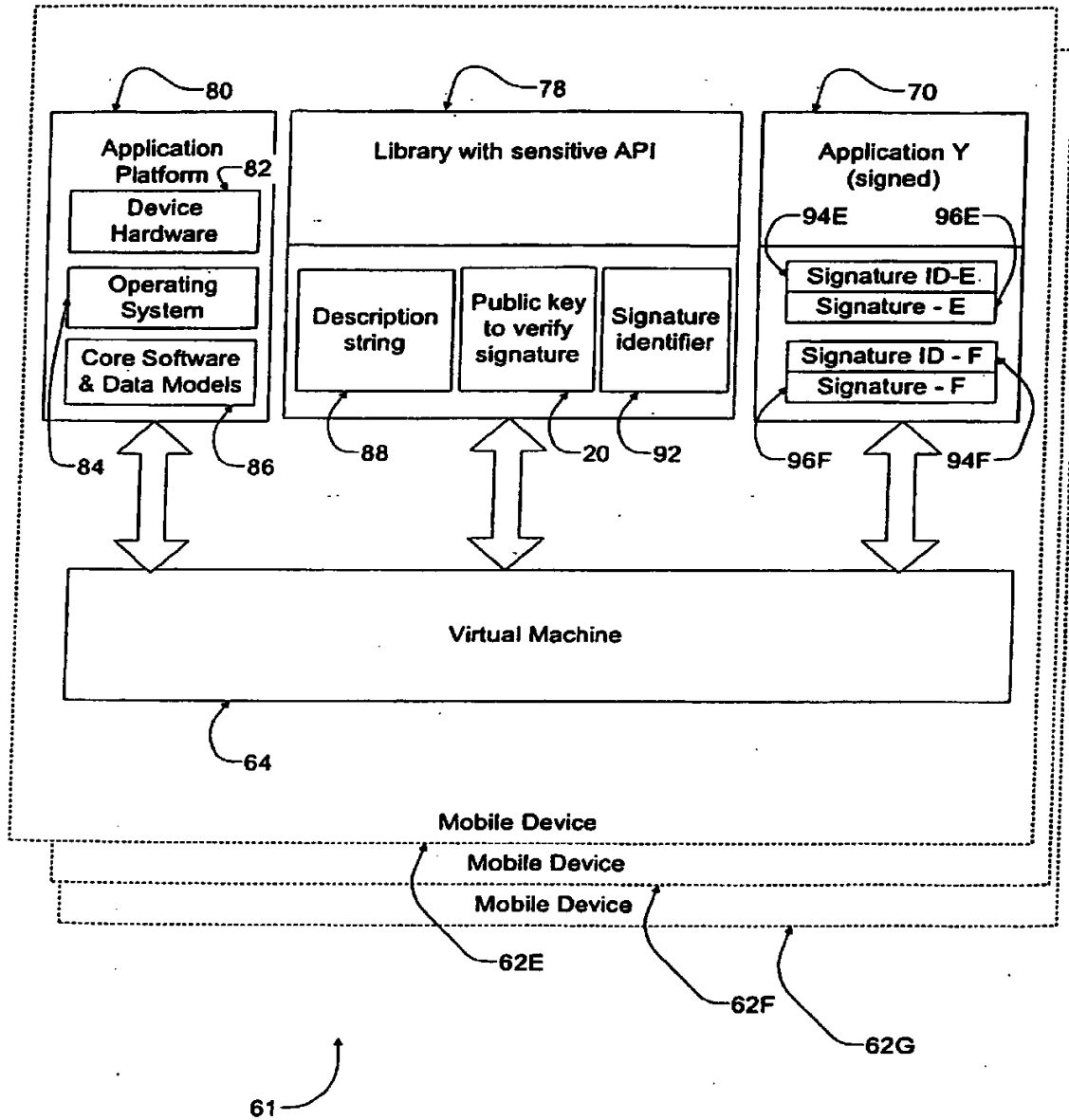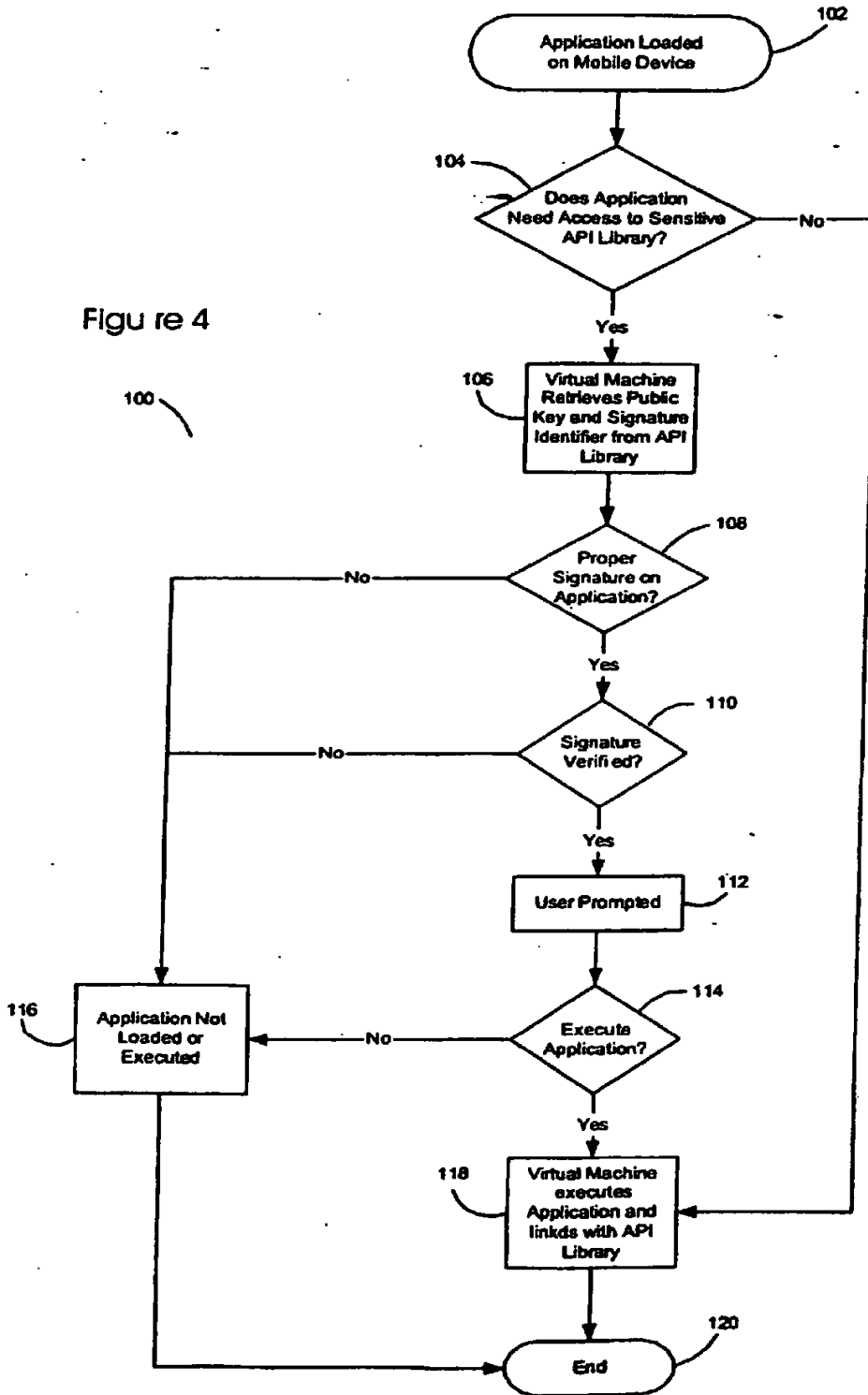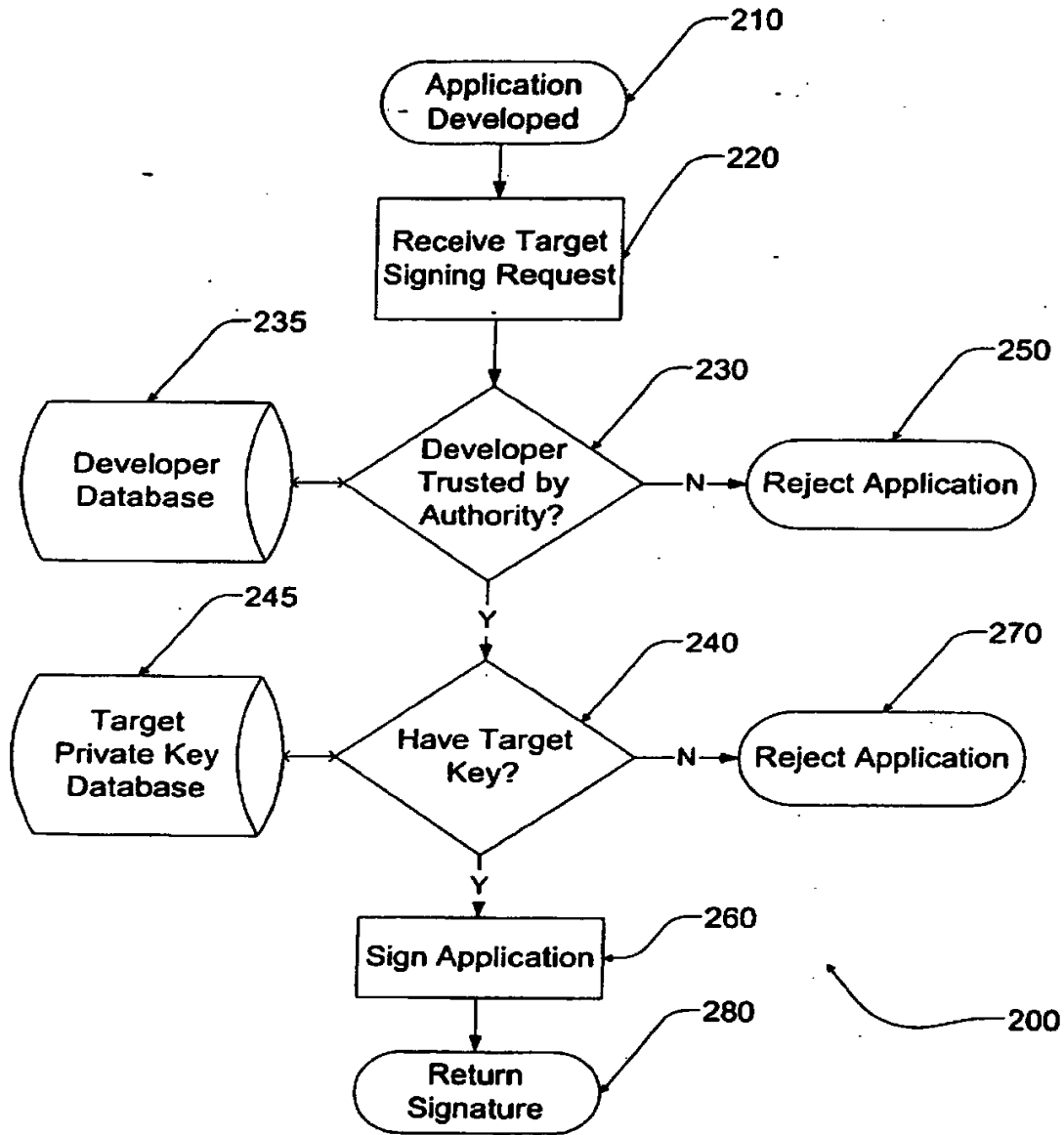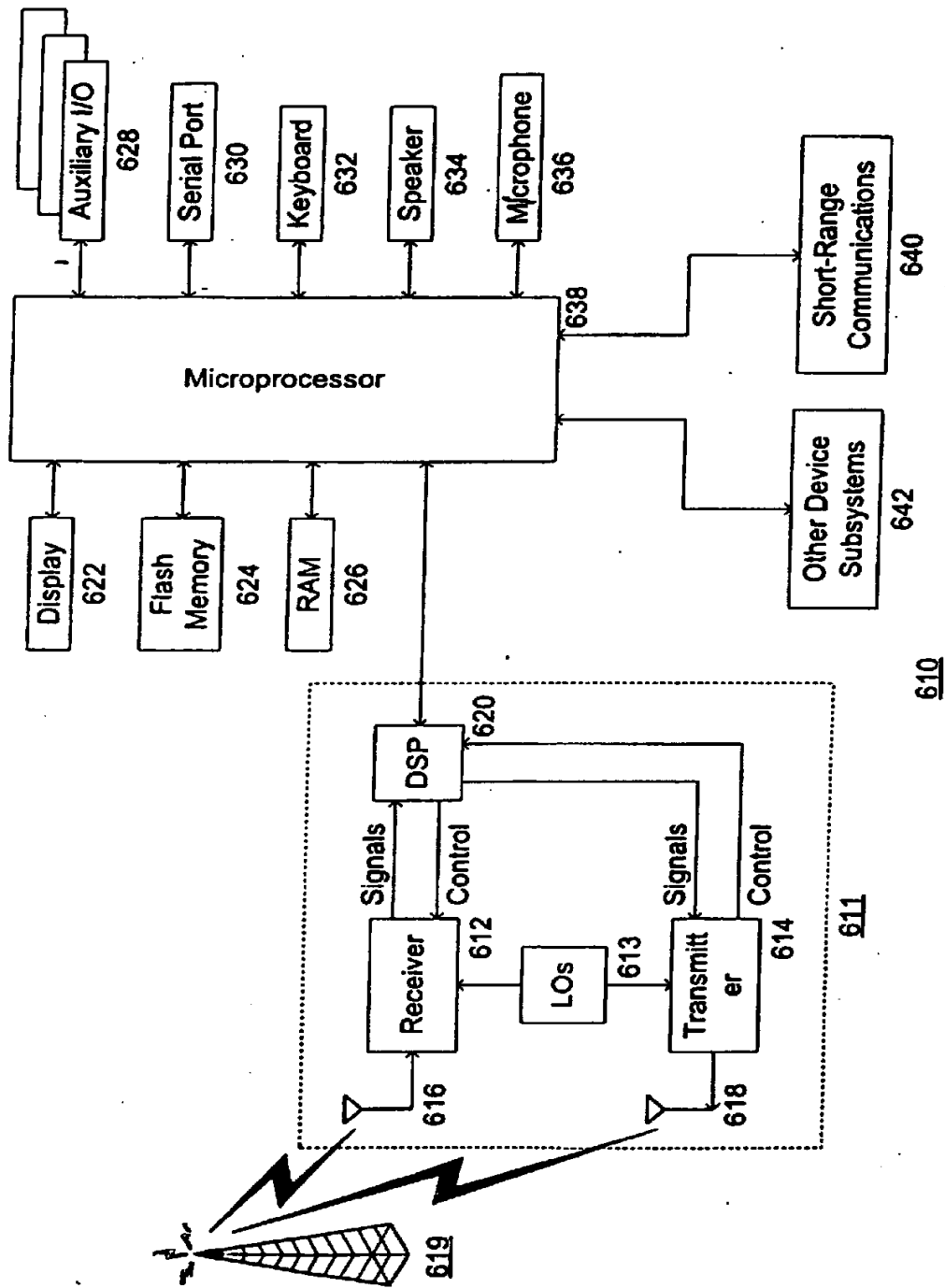
Figure 3

Figure 3A

Figure 4

Application Loaded on Mobile Device — 102

104 — Does Application Need Access to Sensitive API Library?

No

Yes

106 — Virtual Machine Retrieves Public Key and Signature Identifier from API Library

108 — Proper Signature on Application?

No

Yes

110 — Signature Verified?

No

Yes

112 — User Prompted

114 — Execute Application?

No

116 — Application Not Loaded or Executed

Yes

118 — Virtual Machine executes Application and linkds with API Library

120 — End

100

Figure 5

Figure 6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- Handbuch der Chipkarten. 1999, vol. 3 **[0003]**

Europäisches
Patentamt
European
Patent Office
Office européen
des brevets

(11)   **EP 1 626 326 B1**

(12)   **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
01.09.2010 Bulletin 2010/35

(51) Int Cl.:
G06F 1/00 (2006.01)

(21) Application number: **05024663.6**

(22) Date of filing: **20.09.2001**

(54) **SOFTWARE CODE SIGNING SYSTEM AND METHOD**

SYSTEM UND VERFAHREN ZUM UNTERSCHREIBEN EINES SOFTWARE-KODES

SYSTEME ET PROCEDE DE SIGNATURE PAR CODE

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**

(30) Priority: **21.09.2000 US 234152 P
26.09.2000 US 235354 P
20.02.2001 US 270663 P**

(43) Date of publication of application:
15.02.2006 Bulletin 2006/07

(62) Document number(s) of the earlier application(s) in
accordance with Art. 76 EPC:
01973901.0 / 1 320 795

(73) Proprietor: **RESEARCH IN MOTION LIMITED
Waterloo, Ontario N2L 3W8 (CA)**

(72) Inventors:
• **Yach, David P.
Waterloo,
Ontario N2K 2N1 (CA)**

• **Brown, Michael S.
Heidelberg,
Ontario N0B 1Y0 (CA)**
• **Little, Herbert A.
Waterloo,
Ontario N2T 2V8 (CA)**

(74) Representative: **MERH-IP
Matias Erny Reichl Hoffmann
Paul-Heyse-Strasse 29
80336 München (DE)**

(56) References cited:
**EP-A- 0 930 793   WO-A-99/05600
US-A- 5 978 484**

• **"Excerpts ED - RANKL W; EFFING W"
HANDBUCH DER CHIPKARTEN. AUFBAU -
FUNKTIONSWEISE - EINSATZ VON SMART
CARDS, MUENCHEN : CARL HANSER VERLAG,
DE, 1 January 1999 (1999-01-01), pages
197-203,261, XP007908384 ISBN:
978-3-446-21115-5**

EP 1 626 326 B1

Description

BACKGROUND

1. Field of the Invention

[0001] This invention relates generally to the field of security protocols for software applications. More particularly, the invention provides a code signing system and method that is particularly well suited for Java(TM) applications for mobile communication devices, such as Personal Digital Assistants, cellular telephones, and wireless two-way communication devices (collectively referred to hereinafter as "mobile devices" or simply "devices").

2. Description of the Related Art

[0002] Security protocols involving software code signing schemes are known. Typically, such security protocols are used to ensure the reliability of software applications that are downloaded from the Internet. In a typical software code signing scheme, a digital signature is attached to a software application that identifies the software developer. Once the software is downloaded by a user, the user typically must use his or her judgment to determine whether or not the software application is reliable, based solely on his or her knowledge of the software developer's reputation. This type of code signing scheme does not ensure that a software application written by a third party for a mobile device will properly interact with the device's native applications and other resources. Because typical code signing protocols are not secure and rely solely on the judgment of the user, there is a serious risk that destructive, "Trojan horse" type software applications may be downloaded and installed onto a mobile device.

[0003] The disclosure "Handbuch der Chipkarten", W.Rankl/W.Effing, 3. edition, 1999, describes access control via methods to securely load muliple applets in a javacard framework System.

[0004] There also remains a need for network operators to have a system and method to maintain control over which software applications are activated on mobile devices.

[0005] There remains a further need in 2.5G and 3G networks where corporate clients or network operators would like to control the types of software on the devices issued to its employees.

SUMMARY

[0006] A code signing system and method is provided. The code signing system operates in conjunction with a software application having a digital signature and includes an application platform, an application programming interface (API), and a virtual machine. The API is configured to link the software application with the appli-

cation platform. The virtual machine verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0007] A code signing system for operation in conjunction with a software application having a digital signature, according to another embodiment of the invention comprises an application platform, a plurality of APIs, each configured to link the software application with a resource on the application platform, and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0008] According to a further embodiment of the invention, a method of controlling access to sensitive application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to a sensitive API, determining whether or not the software application includes a digital signature associated with the sensitive API, and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0009] In another embodiment of the invention, a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer comprises the steps of receiving the software application from the software developer, reviewing the software application to determine if it may access the API, if the software application may access the API, then appending a digital signature to the software application, verifying the authenticity of a digital signature appended to a software application, and providing access to the API to software applications for which the appended digital signature is authentic.

[0010] A method of restricting access to a sensitive API on a mobile device, according to a further embodiment of the invention, comprises the steps of registering one or more software developers that are trusted to design software applications which access the sensitive API, receiving a hash of a software application, determining if the software application was designed by one of the registered software developers, and if the software application was designed by one of the registered software developers, then generating a digital signature using the hash of the software application, wherein the digital signature may be appended to the software application, and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0011] In a still further embodiment, a method of restricting access to application programming interfaces on a mobile device comprises the steps of loading a software application on the mobile device that requires access to one or more API, determining whether or not the software application includes a digital signature associ-

ated with the mobile device, and if the software application does not include a digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012]

FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention;
FIG. 2 is a flow diagram of the code signing protocol described above with reference to FIG. 1;
FIG. 3 is a block diagram of a code signing system on a mobile device;
FIG. 3A is a block diagram of a code signing system on a plurality of mobile devices;
FIG. 4 is a flow diagram illustrating the operation of the code signing system described above with reference to FIG. 3 and FIG. 3A;
FIG. 5 is a flow diagram illustrating the management of the code signing authorities described with reference to FIG. 3A; and
FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented.

DETAILED DESCRIPTION

[0013]   Referring now to the drawing figures, FIG. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention. An application developer 12 creates a software application 14 (application Y) for a mobile device that requires access to one or more sensitive APIs on the mobile device. The software application Y 14 may, for example, be a Java application that operates on a Java virtual machine installed on the mobile device. An API enables the software application Y to interface with an application platform that may include, for example, resources such as the device hardware, operating system and core software and data models. In order to make function calls to or otherwise interact with such device resources, a software application Y must access one or more APIs. APIs can thereby effectively "bridge" a software application and associated device resources. In this description and the appended claims, references to API access should be interpreted to include access of an API in such a way as to allow a software application Y to interact with one or more corresponding device resources. Providing access to any API therefore allows a software application Y to interact with associated device resources, whereas denying access to an API prevents the software application Y from interacting with the associated resources. For example, a database API may communicate with a device file or data storage system, and access to the database API would provide for interaction between a software application Y and the file or data storage system. A user interface (UI) API would

communicate with controllers and/or control software for such device components as a screen, a keyboard, and any other device components that provide output to a user or accept input from a user. In a mobile device, a radio API may also be provided as an interface to wireless communication resources such as a transmitter and receiver. Similarly, a cryptographic API may be provided to interact with a crypto module which implements crypto algorithms on a device. These are merely illustrative examples of APIs that may be provided on a device. A device may include any of these example APIs, or different APIs instead of or in addition to those described above.
[0014]   Preferably, any API may be classified as sensitive by a mobile device manufacturer, or possibly by an API author, a wireless network operator, a device owner or operator, or some other entity that may be affected by a virus or malicious code in a device software application. For instance, a mobile device manufacturer may classify as sensitive those APIs that interface with cryptographic routines, wireless communication functions, or proprietary data models such as address book or calendar entries. To protect against unauthorized access to these sensitive APIs, the application developer 12 is required to obtain one or more digital signatures from the mobile device manufacturer or other entity that classified any APIs as sensitive, or from a code signing authority 16 acting on behalf of the manufacturer or other entity with an interest in protecting access to sensitive device APIs, and append the signature(s) to the software application Y 14.
[0015]   In one embodiment, a digital signature is obtained for each sensitive API or library that includes a sensitive API to which the software application requires access. In some cases, multiple signatures are desirable. This would allow a service provider, company or network operator to restrict some or all software applications loaded or updated onto a particular set of mobile devices. In this multiple- signature scenario, all APIs are restricted and locked until a "global" signature is verified for a software application. For example, a company may wish to prevent its employees from executing any software applications onto their devices without first obtaining permission from a corporate information technology (IT) or computer services department. All such corporate mobile devices may then be configured to require verification of at least a global signature before a software application can be executed. Access to sensitive device APIs and libraries, if any, could then be further restricted, dependent upon verification of respective corresponding digital signatures.
[0016]   The binary executable representation of software application Y 14 may be independent of the particular type of mobile device or model of a mobile device. Software application Y 14 may for example be in a write-once-run-anywhere binary format such as is the case with Java software applications. However, it may be desirable to have a digital signature for each mobile device type or model, or alternatively for each mobile device

platform or manufacturer. Therefore, software application Y 14 may be submitted to several code signing authorities if software application Y 14 targets several mobile devices.

[0017] Software application Y 14 is sent from the application developer 12 to the code signing authority 16. In the embodiment shown in FIG. 1, the code signing authority 16 reviews the software application Y 14, although as described in further detail below, it is contemplated that the code signing authority 16 may also or instead consider the identity of the application developer 12 to determine whether or not the software application Y 14 should be signed. The code signing authority 16 is preferably one or more representatives from the mobile device manufacturer, the authors of any sensitive APIs, or possibly others that have knowledge of the operation of the sensitive APIs to which the software application needs access.

[0018] If the code signing authority 16 determines that software application Y 14 may access the sensitive API and therefore should be signed, then a signature (not shown) for the software application Y 14 is generated by the code signing authority 16 and appended to the software application Y 14. The signed software application Y 22, comprising the software application Y 14 and the digital signature, is then returned to the application developer 12. The digital signature is preferably a tag that is generated using a private signature key 18 maintained solely by the code signing authority 16. For example, according to one signature scheme, a hash of the software application Y 14 may be generated, using a hashing algorithm such as the Secure Hash Algorithm SHA1, and then used with the private signature key 18 to create the digital signature. In some signature schemes, the private signature key is used to encrypt a hash of information to be signed, such as software application Y 14, whereas in other schemes, the private key may be used in other ways to generate a signature from the information to be signed or a transformed version of the information.

[0019] The signed software application Y 22 may then be sent to a mobile device 28 or downloaded by the mobile device 28 over a wireless network 24. It should be understood, however, that a code signing protocol according to the present invention is not limited to software applications that are downloaded over a wireless network. For instance, in alternative embodiments, the signed software application Y 22 may be downloaded to a personal computer via a computer network and loaded to the mobile device through a serial link, or may be acquired from the application developer 12 in any other manner and loaded onto the mobile device. Once the signed software application Y 22 is loaded on the mobile device 28, each digital signature is preferably verified with a public signature key 20 before the software application Y 14 is granted access to a sensitive API library. Although the signed software application Y 22 is loaded onto a device, it should be appreciated that the software application that may eventually be executed on the device is the software application Y 14. As described above, the signed software application Y 22 includes the software application Y 14 and one or more appended digital signatures (not shown). When the signatures are verified, the software application Y 14 can be executed on the device and access any APIs for which corresponding signatures have been verified.

[0020] The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority 16, and is preferably installed on the mobile device along with the sensitive API. However, the public key 10 may instead be obtained from a public key repository (not shown), using the device 28 or possibly a personal computer system, and installed on the device 28 as needed. According to one embodiment of a signature scheme, the mobile device 28 calculates a hash of the software application Y 14 in the signed software application Y 22, using the same hashing algorithm as the code signing authority 16, and uses the digital signature and the public signature key 20 to recover the hash calculated by the signing authority 16. The resultant locally calculated hash and the hash recovered from the digital signature are then compared, and if the hashes are the same, the signature is verified. The software application Y 14 can then be executed on the device 28 and access any sensitive APIs for which the corresponding signature(s) have been verified. As described above, the invention is in no way limited to this particular illustrative example signature scheme. Other signature schemes, including further public key signature schemes, may also be used in conjunction with the code signing methods and systems described herein.

[0021] FIG. 2 is a flow diagram 30 of the code signing protocol described above with reference to FIG. 1. The protocol begins at step 32. At step 34, a software developer writes the software application Y for a mobile device that requires access to a sensitive API or library that exposes a sensitive API (API library A). As discussed above, some or all APIs on a mobile device may be classified as sensitive, thus requiring verification of a digital signature for access by any software application such as software application Y. In step 36, application Y is tested by the software developer, preferably using a device simulator in which the digital signature verification function has been disabled. In this manner, the software developer may debug the software application Y before the digital signature is acquired from the code signing authority. Once the software application Y has been written and debugged, it is forwarded to the code signing authority in step 38.

[0022] In steps 40 and 42, the code signing authority reviews the software application Y to determine whether or not it should be given access to the sensitive API, and either accepts or rejects the software application. The code signing authority may apply a number of criteria to determine whether or not to grant the software application access to the sensitive API including, for example, the size of the software application, the device resources

accessed by the API, the perceived utility of the software application, the interaction with other software applications, the inclusion of a virus or other destructive code, and whether or not the developer has a contractual obligation or other business arrangement with the mobile device manufacturer. Further details of managing code signing authorities and developers are described below in reference to FIG. 5.

[0023] If the code signing authority accepts the software application Y, then a digital signature, and preferably a signature identification, are appended to the software application Y in step 46. As described above, the digital signature may be generated by using a hash of the software application Y and a private signature key 18. The signature identification is described below with reference to FIGS. 3 and 4. Once the digital signature and signature identification are appended to the software application Y to generate a signed software application, the signed software application Y is returned to the software developer in step 48. The software developer may then license the signed software application Y to be loaded onto a mobile device (step 50). If the code signing authority rejects the software application Y, however, then a rejection notification is preferably sent to the software developer (step 44), and the software application Y will be unable to access any API(s) associated with the signature.

[0024] In an alternative embodiment, the software developer may provide the code signing authority with only a hash of the software application Y, or provide the software application Y in some type of abridged format. If the software application Y is a Java application, then the device independent binary *.class files may be used in the hashing operation, although device dependent files such as *.cod files used by the assignee of the present application may instead be used in hashing or other digital signature operations when software applications are intended for operation on particular devices or device types. By providing only a hash or abridged version of the software application Y, the software developer may have the software application Y signed without revealing proprietary code to the code signing authority. The hash of the software application Y, along with the private signature key 18, may then be used by the code signing authority to generate the digital signature. If an otherwise abridged version of the software application Y is sent to the code signing authority, then the abridged version may similarly be used to generate the digital signature, provided that the abridging scheme or algorithm, like a hashing algorithm, generates different outputs for different inputs. This ensures that every software application will have a different abridged version and thus a different signature that can only be verified when appended to the particular corresponding software application from which the abridged version was generated. Because this embodiment does not enable the code signing authority to thoroughly review the software application for viruses or other destructive code, however, a registration process

between the software developer and the code signing authority may also be required. For instance, the code signing authority may agree in advance to provide a trusted software developer access to a limited set of sensitive APIs.

[0025] In still another alternative embodiment, a software application Y may be submitted to more than one signing authority. Each signing authority may for example be responsible for signing software applications for particular sensitive APIs or APIs on a particular model of mobile device or set of mobile devices that supports the sensitive APIs required by a software application. A manufacturer, mobile communication network operator, service provider, or corporate client for example may thereby have signing authority over the use of sensitive APIs for their particular mobile device model(s), or the mobile devices operating on a particular network, subscribing to one or more particular services, or distributed to corporate employees. A signed software application may then include a software application and at least one appended digital signature appended from each of the signing authorities. Even though these signing authorities in this example would be generating a signature for the same software application, different signing and signature verification schemes may be associated with the different signing authorities.

[0026] FIG. 3 is a block diagram of a code signing system 60 on a mobile device 62. The system 60 includes a virtual machine 64, a plurality of software applications 66-70, a plurality of API libraries 72-78, and an application platform 80. The application platform 80 preferably includes all of the resources on the mobile device 62 that may be accessed by the software applications 66-70. For instance, the application platform may include device hardware 82, the mobile device's operating system 84, or core software and data models 86. Each API library 72-78 preferably includes a plurality of APIs that interface with a resource available in the application platform. For instance, one API library might include all of the APIs that interface with a calendar program and calendar entry data models. Another API library might include all of the APIs that interface with the transmission circuitry and functions of the mobile device 62. Yet another API library might include all of the APIs capable of interfacing with lower-level services performed by the mobile device's operating system 84. In addition, the plurality of API libraries 72-78 may include both libraries that expose a sensitive API 74 and 78, such as an interface to a cryptographic function, and libraries 72 and 76, that may be accessed without exposing sensitive APIs. Similarly, the plurality of software applications 66-70 may include both signed software applications 66 and 70 that require access to one or more sensitive APIs, and unsigned software applications such as 68. The virtual machine 64 is preferably an object oriented run-time environment such as Sun Micro System's J2ME(TM) (Java 2 Platform, Micro Edition), which manages the execution of all of the software applications 66-70 operating on the mobile de-

vice 62, and links the software applications 66-70 to the various API libraries 72-78.

[0027] Software application Y 70 is an example of a signed software application. Each signed software application preferably includes an actual software application such as software application Y comprising for example software code that can be executed on the application platform 80, one or more signature identifications 94 and one or more corresponding digital signatures 96. Preferably each digital signature 96 and associated signature identification 94 in a signed software application 66 or 70 corresponds to a sensitive API library 74 or 78 to which the software application X or software application Y requires access. The sensitive API library 74 or 78 may include one or more sensitive APIs. In an alternative embodiment, the signed software applications 66 and 70 may include a digital signature 96 for each sensitive API within an API library 74 or 78. The signature identifications 94 may be unique integers or some other means of relating a digital signature 96 to a specific API library 74 or 78, API, application platform 80, or model of mobile device 62.

[0028] API library A 78 is an example of an API library that exposes a sensitive API. Each API library 74 and 78 including a sensitive API should preferably include a description string 88, a public signature key 20, and a signature identifier 92. The signature identifier 92 preferably corresponds to a signature identification 94 in a signed software application 66 or 70, and enables the virtual machine 64 to quickly match a digital signature 96 with an API library 74 or 78. The public signature key 20 corresponds to the private signature key 18 maintained by the code signing authority, and is used to verify the authenticity of a digital signature 96. The description string 88 may for example be a textual message that is displayed on the mobile device when a signed software application 66 or 70 is loaded, or alternatively when a software application X or Y attempts to access a sensitive API.

[0029] Operationally, when a signed software application 68-70, respectively including a software application X, Z, or Y, that requires access to a sensitive API library 74 or 78 is loaded onto a mobile device, the virtual machine 64 searches the signed for an appended digital signature 96 associated with the API library 74 or 78. Preferably, the appropriate digital signature 96 is located by the virtual machine 64 by matching the signature identifier 92 in the API library 74 or 78 with a signature identification 94 on the signed software application. If the signed software application includes the appropriate digital signature 96, then the virtual machine 64 verifies its authenticity using the public signature key 20. Then, once the appropriate digital signature 96 has been located and verified, the description string 88 is preferably displayed on the mobile device before the software application X or Y is executed and accesses the sensitive API. For instance, the description string 88 may display a message stating that "Application Y is attempting to access

API Library A," and thereby provide the mobile device user with the final control to grant or deny access to the sensitive API.

[0030] FIG. 3A is a block diagram of a code signing system 61 on a plurality of mobile devices 62E, 62F and 62G. The system 61 includes a plurality of mobile devices each of which only three are illustrated, mobile devices 62E, 62F and 62G. Also shown is a signed software application 70, including a software application Y to which two digital signatures 96E and 96F with corresponding signature identifications 94E and 94F have been appended. In the example system 61, each pair composed of a digital signature and identification, 94E/96E and 94F/96F, corresponds to a model of mobile device 62, API library 78, or associated platform 80. If signature identifications 94E and 94F correspond to different models of mobile device 62, then when a signed software application 70 which includes a software application Y that requires access to a sensitive API library 78 is loaded onto mobile device 62E, the virtual machine 64 searches the signed software application 70 for a digital signature 96E associated with the API library 78 by matching identifier 94E with signature identifier 92. Similarly, when a signed software application 70 including a software application Y that requires access to a sensitive API library 78 is loaded onto a mobile device 62F, the virtual machine 64 in device 62F searches the signed software application 70 for a digital signature 96F associated with the API library 78. However, when a software application Y in a signed software application 70 that requires access to a sensitive API library 78 is loaded onto a mobile device model for which the application developer has not obtained a digital signature, device 62G in the example of FIG. 3A, the virtual machine 64 in the device 64G does not find a digital signature appended to the software application Y and consequently, access to the API library 78 is denied on device 62G. It should be appreciated from the foregoing description that a software application such as software application Y may have multiple device-specific, library-specific, or API-specific signatures or some combination of such signatures appended thereto. Similarly, different signature verification requirements may be configured for the different devices. For example, device 62E may require verification of both a global signature, as well as additional signatures for any sensitive APIs to which a software application, requires access in order for the software application to be executed, whereas device 62F may require verification of only a global signature and device 62G may require verification of signatures only for its sensitive APIs. It should also be apparent that a communication system may include devices (not shown) on which a software application Y received as part of a signed software application such as 70 may execute without any signature verification. Although a signed software application has one or more signatures appended thereto, the software application Y might possibly be executed on some devices without first having any of its signature(s) verified. Signing of a software ap-

plication preferably does not interfere with its execution on devices in which digital signature verification is not implemented.

[0031] FIG. 4 is a flow diagram 100 illustrating the operation of the code signing system described above with reference to FIGS. 3 and 3A. In step 102, a software application is loaded onto a mobile device. Once the software application is loaded, the device, preferably using a virtual machine, determines whether or not the software application requires access to any API libraries that expose a sensitive API (step 104). If not, then the software application is linked with all of its required API libraries and executed (step 118). If the software application does require access to a sensitive API, however, then the virtual machine verifies that the software application includes a valid digital signature associated any sensitive APIs to which access is required, in steps 106-116.

[0032] In step 106, the virtual machine retrieves the public signature key 20 and signature identifier 92 from the sensitive API library. The signature identifier 92 is then used by the virtual machine in step 108 to determine whether or not the software application has an appended digital signature 96 with a corresponding signature identification 94. If not, then the software application has not been approved for access to the sensitive API by a code signing authority, and the software application is preferably prevented from being executed in step 116. In alternative embodiments, a software application without a proper digital signature 96 may be purged from the mobile device, or may be denied access to the API library exposing the sensitive API but executed to the extent possible without access to the API library. It is also contemplated that a user may be prompted for an input when signature verification fails, thereby providing for user control of such subsequent operations as purging of the software application from the device.

[0033] If a digital signature 96 corresponding to the sensitive API library is appended to the software application and located by the virtual machine, then the virtual machine uses the public key 20 to verify the authenticity of the digital signature 96 in step 110. This step may be performed, for example, by using the signature verification scheme described above or other alternative signature schemes. If the digital signature 96 is not authentic, then the software application is preferably either not executed, purged, or restricted from accessing the sensitive API as described above with reference to step 116. If the digital signature is authentic, however, then the description string 88 is preferably displayed in step 112, warning the mobile device user that the software application requires access to a sensitive API, and possibly prompting the user for authorization to execute or load the software application (step 114). When more than one signature is to be verified for a software application, then the steps 104-110 are preferably repeated for each signature before the user is prompted in step 112. If the mobile device user in step 114 authorizes the software application, then it may be executed and linked to the sensitive API library

in step 118.

[0034] FIG. 5 is a flow diagram 200 illustrating the management of the code signing authorities described with reference to FIG. 3A. At step 210, an application developer has developed a new software application which is intended to be executable one or more target device models or types. The target devices may include sets of devices from different manufacturers, sets of device models or types from the same manufacturer, or generally any sets of devices having particular signature and verification requirements. The term "target device" refers to any such set of devices having a common signature requirement. For example, a set of devices requiring verification of a device-specific global signature for execution of all software applications may comprise a target device, and devices that require both a global signature and further signatures for sensitive APIs may be part of more than one target device set. The software application may be written in a device independent manner by using at least one known API, supported on at least one target device with an API library. Preferably, the developed software application is intended to be executable on several target devices, each of which has its own at least one API library.

[0035] At step 220, a code signing authority for one target device receives a target-signing request from the developer. The target signing request includes the software application or a hash of the software application, a developer identifier, as well as at least one target device identifier which identifies the target device for which a signature is being requested. At step 230, the signing authority consults a developer database 235 or other records to determine whether or not to trust developer 220. This determination can be made according to several criteria discussed above, such as whether or not the developer has a contractual obligation or has entered into some other type of business arrangement with a device manufacturer, network operator, service provider, or device manufacturer. If the developer is trusted, then the method proceeds at step 240. However, if the developer is not trusted, then the software application is rejected (250) and not signed by the signing authority. Assuming the developer was trusted, at step 240 the signing authority determines if it has the target private key corresponding to the submitted target identifier by consulting a private key store such as a target private key database 245. If the target private key is found, then a digital signature for the software application is generated at step 260 and the digital signature or a signed software application including the digital signature appended to the software application is returned to the developer at step 280. However, if the target private key is not found at step 240, then the software application is rejected at step 270 and no digital signature is generated for the software application.

[0036] Advantageously, if target signing authorities follow compatible embodiments of the method outlined in FIG. 5, a network of target signing authorities may be established in order to expediently manage code signing

authorities and a developer community code signing process providing signed software applications for multiple targets with low likelihood of destructive code.

[0037] Should any destructive or otherwise problematic code be found in a software application or suspected because of behavior exhibited when a software application is executed on a device, then the registration or privileges of the corresponding application developer with any or all signing authorities may also be suspended or revoked, since the digital signature provides an audit trail through which the developer of a problematic software application may be identified. In such an event, devices may be informed of the revocation by being configured to periodically download signature revocation lists, for example. If software applications for which the corresponding digital signatures have been revoked are running on a device, the device may then halt execution of any such software application and possibly purge the software application from its local storage. If preferred, devices may also be configured to re-execute digital signature verifications, for instance periodically or when a new revocation list is downloaded.

[0038] Although a digital signature generated by a signing authority is dependent upon authentication of the application developer and confirmation that the application developer has been properly registered, the digital signature is preferably generated from a hash or otherwise transformed version of the software application and is therefore application-specific. This contrasts with known code signing schemes, in which API access is granted to any software applications arriving from trusted application developers or authors. In the code signing systems and methods described herein, API access is granted on an application-by-application basis and thus can be more strictly controlled or regulated.

[0039] FIG. 6 is a block diagram of a mobile communication device in which a code signing system and method may be implemented. The mobile communication device 610 is preferably a two-way communication device having at least voice and data communication capabilities. The device preferably has the capability to communicate with other computer systems on the Internet. Depending on the functionality provided by the device, the device may be referred to as a data messaging device, a two-way pager, a cellular telephone with data messaging capabilities, a wireless Internet appliance or a data communication device (with or without telephony capabilities).

[0040] Where the device 610 is enabled for two-way communications, the device will incorporate a communication subsystem 611, including a receiver 612, a transmitter 614, and associated components such as one or more, preferably embedded or internal, antenna elements 616 and 618, local oscillators (LOs) 613, and a processing module such as a digital signal processor (DSP) 620. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem 611 will be dependent upon the com-

munication network in which the device is intended to operate. For example, a device 610 destined for a North American market may include a communication subsystem 611 designed to operate within the Mobitex(TM) mobile communication system or DataTAC(TM) mobile communication system, whereas a device 610 intended for use in Europe may incorporate a General Packet Radio Service (GPRS) communication subsystem 611.

[0041] Network access requirements will also vary depending upon the type of network 919. For example, in the Mobitex and DataTAC networks, mobile devices such as 610 are registered on the network using a unique identification number associated with each device. In GPRS networks however, network access is associated with a subscriber or user of a device 610. A GPRS device therefore requires a subscriber identity module (not shown), commonly referred to as a SIM card, in order to operate on a GPRS network. Without a SIM card, a GPRS device will not be fully functional. Local or non-network communication functions (if any) may be operable, but the device 610 will be unable to carry out any functions involving communications over network 619, other than any legally required operations such as "911 " emergency calling.

[0042] When required network registration or activation procedures have been completed, a device 610 may send and receive communication signals over the network 619. Signals received by the antenna 616 through a communication network 619 are input to the receiver 612, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in FIG. 6, analog to digital conversion. Analog to digital conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP 620. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by the DSP 620 and input to the transmitter 614 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission over the communication network 619 via the antenna 618.

[0043] The DSP 620 not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in the receiver 612 and transmitter 614 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 620.

[0044] The device 610 preferably includes a microprocessor 638 which controls the overall operation of the device. Communication functions, including at least data and voice communications, are performed through the communication subsystem 611. The microprocessor 638 also interacts with further device subsystems or resources such as the display 622, flash memory 624, random access memory (RAM) 626, auxiliary input/output (I/O) subsystems 628, serial port 630, keyboard 632, speaker 634, microphone 636, a short-range communications subsystem 640 and any other device subsystems gen-

erally designated as 642. APIs, including sensitive APIs requiring verification of one or more corresponding digital signatures before access is granted, may be provided on the device 610 to interface between software applications and any of the resources shown in FIG. 6.

[0045] Some of the subsystems shown in FIG. 6 perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard 632 and display 622 for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0046] Operating system software used by the microprocessor 638, and possibly APIs to be accessed by software applications, is preferably stored in a persistent store such as flash memory 624, which may instead be a read only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device software applications, or parts thereof, may be temporarily loaded into a volatile store such as RAM 626. It is contemplated that received and transmitted communication signals may also be stored to RAM 626.

[0047] The microprocessor 638, in addition to its operating system functions, preferably enables execution of software applications on the device. A predetermined set of applications which control basic device operations, including at least data and voice communication applications for example, will normally be installed on the device 610 during manufacture. A preferred application that may be loaded onto the device may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the device user such as, but not limited to e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the device to facilitate storage of PIM data items on the device. Such PIM application would preferably have the ability to send and receive data items, via the wireless network. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network, with the device user's corresponding data items stored or associated with a host computer system thereby creating a mirrored host computer on the mobile device with respect to the data items at least. This would be especially advantageous in the case where the host computer system is the mobile device user's office computer system. Further applications, including signed software applications as described above, may also be loaded onto the device 610 through the network 619, an auxiliary I/O subsystem 62S, serial port 630, short-range communications subsystem 640 or any other suitable subsystem 642. The device microprocessor 638 may then verify any digital signatures, possibly including both "global" device signatures and API-specific signatures, appended to such a software application before the soft-

ware application can be executed by the microprocessor 638 and/or access any associated sensitive APIs. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the device 610, through a crypto API and a crypto module which implements crypto algorithms on the device (not shown).

[0048] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem 611 and input to the microprocessor 638, which will preferably further process the received signal for output to the display 622, or alternatively to an auxiliary I/O device 628. A user of device 610 may also compose data items such as email messages for example, using the keyboard 632, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display 622 and possibly an auxiliary I/O device 628. Such composed items may then be transmitted over a communication network through the communication subsystem 611.

[0049] For voice communications, overall operation of the device 610 is substantially similar, except that received signals would preferably be output to a speaker 634 and signals for transmission would be generated by a microphone 636. Alternative voice or audio I/O subsystems such as a voice message recording subsystem may also be implemented on the device 610. Although voice or audio signal output is preferably accomplished primarily through the speaker 634, the display 622 may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information for example.

[0050] The serial port 630 in FIG. 6 would normally be implemented in a personal digital assistant (PDA)-type communication device for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port 630 would enable a user to set preferences through an external device or software application and would extend the capabilities of the device by providing for information or software downloads to the device 610 other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the device through a direct and thus reliable and trusted connection to thereby enable secure device communication.

[0051] A short-range communications subsystem 640 is a further optional component which may provide for communication between the device 624 and different systems or devices, which need not necessarily be similar devices. For example, the subsystem 640 may include an infrared device and associated circuits and components or a Bluetooth(TM) communication module to provide for communication with similarly-enabled sys-

tems and devices.

[0052] The embodiments described herein are examples of structures, systems or methods having elements corresponding to the elements of the invention recited in the claims. This written description may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements of the invention recited in the claims. The intended scope of the invention thus includes other structures, systems or methods that do not differ from the literal language of the claims, and further includes other structures, systems or methods with insubstantial differences from the literal language of the claims.

[0053] For example, when a software application is rejected at step 250 in the method shown in FIG. 5, the signing authority may request that the developer sign a contract or enter into a business relationship with a device manufacturer or other entity on whose behalf the signing authority acts. Similarly, if a software application is rejected at step 270, authority to sign the software application may be delegated to a different signing authority. The signing of a software application following delegation of signing of the software application to the different authority can proceed substantially as shown in FIG. 5, wherein the target signing authority that received the original request from the trusted developer at step 220 requests that the software application be signed by the different signing authority on behalf of the trusted developer from the target signing authority. Once a trust relationship has been established between code signing authorities, target private code signing keys could be shared between code signing authorities to improve performance of the method at step 240, or a device may be configured to validate digital signatures from either of the trusted signing authorities.

[0054] In addition, although described primarily in the context of software applications, code signing systems and methods according to the present invention may also be applied to other device-related components, including but in no way limited to, commands and associated command arguments, and libraries configured to interface with device resources. Such commands and libraries may be sent to mobile devices by device manufacturers, device owners, network operators, service providers, software application developers and the like. It would be desirable to control the execution of any command that may affect device operation, such as a command to change a device identification code or wireless communication network address for example, by requiring verification of one or more digital signatures before a command can be executed on a device, in accordance with the code signing systems and methods described and claimed herein.

[0055] As has been described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; an application programming interface (API) configured to link the software application with the application platform; and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application.

[0056] The virtual machine may deny the software application access to the API if the digital signature is not authentic. The virtual machine may purge the software application if the digital signature is not authentic. The code signing system may be installed on a mobile device. The digital signature may be generated by a code signing authority.

[0057] The code signing system may further comprise a plurality of API libraries each including a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

[0058] One or more of the plurality of API libraries may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive API libraries by the software application. The software application may include a unique digital signature for each sensitive API library. The software application may include a signature identification for each unique digital signature; each sensitive API library may include a signature identifier; and the virtual machine may compare the signature identification and the signature identifier to match the unique digital signatures with sensitive API libraries.

[0059] The digital signature may be generated using a private signature key, and the virtual machine may use a public signature key to verify the authenticity of the digital signature. The digital signature may be generated by applying the private signature key to a hash of the software application; and the virtual machine may verify the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

[0060] The API may further comprise a description string that is displayed by the mobile device when the software application attempts to access the API. The application platform may comprise an operating system. The application platform may comprise one or more core functions of a mobile device. The application platform may comprise hardware on a mobile device. The hardware may comprise a subscriber identity module (SIM) card. The software application may be a Java application for a mobile device. The API may interface with a cryptographic routine on the application platform. The API may interface with a proprietary data model on the application platform. The virtual machine may be a Java virtual machine installed on a mobile device.

[0061] As also described, a code signing system for operation in conjunction with a software application having a digital signature, comprises an application platform; a plurality of application programming interfaces (APIs), each configured to link the software application with a resource on the application platform; and a virtual machine that verifies the authenticity of the digital signature

in order to control access to the API by the software application, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

[0062] The plurality of APIs may be included in an API library. One or more of the plurality of APIs may be classified as sensitive, and the virtual machine may use the digital signature to control access to the sensitive APIs. For operation in conjunction with a plurality of software applications, one or more of the plurality of software applications may have a digital signature, and the virtual machine may verify the authenticity of the digital signature of each of the one or more of the plurality of software applications in order to control access to the sensitive APIs by each of the plurality of software applications. The resource on the application platform may comprise a wireless communication system. The resource on the application platform may comprise a cryptographic module which implements cryptographic algorithms. The resource on the application platform may comprise a data store. The resource on the application platform may comprise a user interface (UI).

[0063] As has also been described, a method of controlling access to sensitive application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to a sensitive application programming interface (API); determining whether or not the software application includes a digital signature associated with the sensitive API; and if the software application does not include a digital signature associated with the sensitive API, then denying the software application access to the sensitive API.

[0064] The method my comprise the additional step of: if the software application does not include a digital signature associated with the sensitive API, then purging the software application from the mobile device. The digital signature may be generated by a code signing authority. The method may comprise the additional steps of: if the software application includes a digital signature associated with the sensitive API, then verifying the authenticity of the digital signature; and if the digital signature is not authentic, then denying the software application access to the sensitive API. The method may further comprise the additional step of: if the digital signature is not authentic, then purging the software application from the mobile device. The digital signature may be generated by applying a private signature key to a hash of the software application, and the step of verifying the authenticity of the digital signature may be performed by a method comprising the steps of: storing a public signature key that corresponds to the private signature key on the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash. The digital signature may be generated by calculating a hash of the software application and applying the

private signature key. The method may comprise the additional step of: displaying a description string that notifies a user of the mobile device that the software application requires access to the sensitive API. The method may further comprise the additional step of: receiving a command from the user granting or denying the software application access to the sensitive API.

[0065] Further has been described a method of controlling access to an application programming interface (API) on a mobile device by a software application created by a software developer, comprising the steps of: receiving the software application from the software developer; reviewing the software application to determine if it may access the API; if the software application may access the API, then appending a digital signature to the software application; verifying the authenticity of a digital signature appended to a software application; and providing access to the API to software applications for which the appended digital signature is authentic.

[0066] The step of reviewing the software application may be performed by a code signing authority. The step of appending the digital signature to the software application may be performed by a method comprising the steps of: calculating a hash of the software application; and applying a signature key to the hash of the software application to generate the digital signature. The hash of the software application may be calculated using the Secure Hash Algorithm (SHA1). The step of verifying the authenticity of a digital signature may comprise the steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; and determining if the digital signature is authentic by comparing the calculated hash with the recovered hash. The method may further comprise the step of, if the digital signature is not authentic, then denying the software application access to the API. The signature key may be a private signature key and the corresponding signature key is a public signature key.

[0067] Also has been described, a method of controlling access to a sensitive application programming interface (API) on a mobile device, comprising the steps of: registering one or more software developers that are trusted to design software applications which access the sensitive API; receiving a hash of a software application; determining if the software application was designed by one of the registered software developers; and if the software application was designed by one of the registered software developers, then generating a digital signature using the hash of the software application, wherein the digital signature may be appended to the software application; and the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

[0068] The step of generating the digital signature may be performed by a code signing authority. The step of generating the digital signature may be performed by ap-

plying a signature key to the hash of the software application. The mobile device may verify the authenticity of the digital signature by performing the additional steps of: providing a corresponding signature key on the mobile device; calculating the hash of the software application on the mobile device to obtain a calculated hash; applying the corresponding signature key to the digital signature to obtain a recovered hash; determining if the digital signature is authentic by comparing the calculated hash with the recovered hash; and if the digital signature is not authentic, then denying the software application access to the sensitive API.

[0069] As has been described, a method of restricting access to application programming interfaces on a mobile device, comprises the steps of: loading a software application on the mobile device that requires access to one or more application programming interface (API); determining whether or not the software application includes an authentic digital signature associated with the mobile device; and if the software application does not include an authentic digital signature associated with the mobile device, then denying the software application access to the one or more APIs.

[0070] The method may comprise the additional step of: if the software application does not include an authentic digital signature associated with the mobile device, then purging the software application from the mobile device. The software application may include a plurality of digital signatures. The plurality of digital signatures may include digital signatures respectively associated with different types of mobile devices.

[0071] Each of the plurality of digital signatures may be generated by a respective corresponding code signing authority. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the additional steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature. The one or more APIs may include one or more APIs classified as sensitive, and the method may further comprise the steps of, for each sensitive API: determining whether or not the software application includes an authentic digital signature associated with the sensitive API; and if the software application does not include an authentic digital signature associated with the sensitive API, then denying the software application access to the sensitive API. Each of the plurality of digital signatures may be generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application. The step of determining whether or not the software application includes an authentic digital signature associated with the mobile device may comprise the steps of: determining if the software application includes a digital signature associated with the mobile device; and if so, then verifying the authenticity of the digital signature, wherein the step of verifying the

authenticity of the digital signature is performed by a method comprising the steps of: storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the signature associated with the mobile device; generating a hash of the software application to obtain a generated hash; applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash.

Claims

1. A method of restricting access to application programming interfaces on a mobile device (62), comprising the steps of:

   loading a software application (66) having a digital signature (96) and a signature identification (94) on the mobile device (62) that requires access to one or more application programming interfaces (APIs) having at least one signature identifier (92),
   authenticating the digital signature (96) where the signature identification (94) corresponds with the signature identifier (92), and
   denying the software application (66) access to the one or more APIs where the software application (66) does not include an authentic digital signature (96).

2. The method of claim 1, wherein the digital signature (96) and signature identification (94) are associated with a type of mobile device (62).

3. The method of claim 1 or 2, comprising the additional step of:

   purging the software application (66) from the mobile device (62) where the software application (66) does not include an authentic digital signature (96).

4. The method of any preceding claim, wherein:

   the software application (66) includes a plurality of digital signatures (96) and signature identifications (94); and
   the plurality of digital signatures (96) and signature identifications includes digital signatures and signature identifications respectively associated with different types of mobile devices (62).

5. The method of claim 4, wherein each of the plurality of digital signatures (96) and associated signature identifications (94) are generated by a respective corresponding code signing authority.

6. The method of any preceding claim, wherein the step of determining whether the software application (66) includes an authentic digital signature (96) comprises the additional steps of:

verifying the authenticity of the digital signature (96) where the signature identification (94) corresponds with respective ones of the at least one signature identifier (92).

7. The method of claim 5, wherein each of the plurality of digital signatures (96) and signature identifications (94) are generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application (66).

8. The method of any preceding claim, wherein the step of authenticating the digital signature (96) where the signature identification (94) corresponds with the signature identifier (92) comprises the steps of:

verifying that the signature identification (94) corresponds with the signature identifier (92) authenticating the digital signature (96) where signature identification (94) corresponds with the signature identifier (92) comprising the steps of:

storing a public signature key on a mobile device (62) that corresponds to the private signature key associated with the code signing authority which generates the digital signature (96);
generating a hash of the software application (66) to obtain a generated hash; applying the public signature key to the digital signature (96) to obtain a recovered hash; and comparing the generated hash with the recovered hash.

9. The method of any preceding claim, wherein:

the mobile device (62) includes a plurality of APIs;
at least one of the plurality of APIs is classified as sensitive;
access to any of the plurality of APIs requires an authentic global signature;
access to each of the plurality of sensitive APIs requires an authentic global signature and an authentic digital signature associated with a signature identification (94);
the step of determining whether the software application (66) includes an authentic digital signature and signature identification comprises the steps of
determining whether the one or more APIs to which the software application (66) requires ac-

cess includes a sensitive API;
determining whether the software application (66) includes an authentic global signature; and
determining whether the software application (66) includes an authentic digital signature and signature identification where the one or more APIs to which the software application (66) requires access includes a sensitive API and the software application (66) includes an authentic global signature; and
the step of denying the software application (66) access to the one or more APIs comprises the steps of:

denying the software application (66) access to the one or more APIs where the software application (66) does not include an authentic global signature; and
denying the software application (66) access to the sensitive API where the one or more APIs to which the software application (66) requires access includes a sensitive API, the software application (66) includes an authentic global signature, and the software application (66) does not include an authentic digital signature and signature identifier required to access the sensitive API.

Patentansprüche

1. Verfahren zum Beschränken eines Zugangs zu Anwendungsprogrammierungsschnittstellen auf einer mobilen Vorrichtung (62), das die Schritte aufweist:

Laden einer Softwareanwendung (66) mit einer digitalen Signatur (96) und
einer Signaturidentifikation (94) auf die mobile Vorrichtung (62), die einen Zugang zu einer oder mehreren Anwendungsprogrammierungs-schnittstellen (APIs - application programming interfaces) mit zumindest einem Signaturidenti-fizierer (92) erfordert,
Authentisieren der digitalen Signatur (96), wenn die Signaturidentifikation (94) dem Signaturi-dentifizierer (92) entspricht, und
Verweigern eines Zugangs zu der einen oder mehreren APIs für die Softwareanwendung (66), wenn die Softwareanwendung (66) keine authentische digitale Signatur (96) umfasst.

2. Verfahren gemäß Anspruch 1, wobei die digitale Signatur (96) und die Signaturidentifikation (94) zu einem Typ einer mobilen Vorrichtung (62) gehören.

3. Verfahren gemäß Anspruch 1 oder 2, das den zu-

sätzlichen Schritt aufweist:

> Löschen der Softwareanwendung (66) aus der mobilen Vorrichtung (62),
> wenn die Softwareanwendung (66) keine authentische digitale Signatur (96) umfasst.

4. Verfahren gemäß einem beliebigen vorangegangenen Anspruch, wobei:

> die Softwareanwendung (66) eine Vielzahl von digitalen Signaturen (96) und Signaturidentifikationen (94) umfasst; und
> die Vielzahl von digitalen Signaturen (96) und Signaturidentifikationen (94) digitale Signaturen und Signaturidentifikationen umfassen, die jeweils zu unterschiedlichen Typen von mobilen Vorrichtungen (62) gehören.

5. Verfahren gemäß Anspruch 4, wobei jede der Vielzahl von digitalen Signaturen (96) und zugehörigen Signaturidentifikationen (94) durch eine jeweilige entsprechende Codesignierautorität erzeugt wird.

6. Verfahren gemäß einem beliebigen vorangegangenen Anspruch, wobei der Schritt des Bestimmens, ob die Softwareanwendung (66) eine authentische digitale Signatur (96) umfasst, die zusätzlichen Schritte aufweist:

> Verifizieren der Authentizität der digitalen Signatur (96), wenn die Signaturidentifikation (94) den jeweiligen des zumindest einen Signaturidentifizierers (92) entspricht.

7. Verfahren gemäß Anspruch 5, wobei jede der Vielzahl von digitalen Signaturen (96) und Signaturidentifikationen (94) durch ihre entsprechenden Codesignierautorität erzeugt wird durch Anwenden eines jeweiligen privaten Signaturschlüssels, der zu der Codesignierautorität gehört, auf einen Hash der Softwareanwendung (66).

8. Verfahren gemäß einem beliebigen vorangegangenen Anspruch, wobei der Schritt des Authentisierens der digitalen Signatur (96), wenn die Signaturidentifikation (94) dem Signaturidentifizierer (92) entspricht, die Schritte aufweist:

> Verifizieren, dass die Signaturidentifikation (94) dem Signaturidentifizierer (92) entspricht, Authentisieren der digitalen Signatur (96), wenn die Signaturidentifikation (94) dem Signaturidentifizierer (92) entspricht, das die Schritte aufweist:

>> Speichern eines öffentlichen Signaturschlüssels auf einer mobilen Vorrichtung

(62), der dem privaten Signaturschlüssel entspricht, der zu der Codesignierautorität gehört, welche die digitale Signatur (96) erzeugt;
Erzeugen eines Hashs der Softwareanwendung (66), um einen erzeugten Hash zu erlangen;
Anwenden des öffentlichen Signaturschlüssels auf die digitale Signatur (96), um einen wiedergewonnenen Hash zu erlangen; und
Vergleichen des erzeugten Hashs mit dem wiedergewonnenen Hash.

9. Verfahren gemäß einem beliebigen vorangegangenen Anspruch, wobei:

> die mobile Vorrichtung (62) eine Vielzahl von APIs umfasst;
> zumindest eine der Vielzahl von APIs als sensitiv klassifiziert ist;
> ein Zugang zu einer der Vielzahl von APIs eine authentische globale Signatur erfordert;
> ein Zugang zu jeder der Vielzahl von sensitiven APIs eine authentische globale Signatur und eine authentische digitale Signatur erfordert, die zu einer Signaturidentifikation (94) gehört;
> der Schritt des Bestimmens, ob die Softwareanwendung (66) eine authentische digitale Signatur und eine Signaturidentifikation umfasst, die Schritte aufweist:

>> Bestimmen, ob die eine oder mehrere APIs, zu der/denen die Softwareanwendung (66) einen Zugang erfordert, eine sensitive API umfasst;
>> Bestimmen, ob die Softwareanwendung (66) eine authentische globale Signatur umfasst; und
>> Bestimmen, ob die Softwareanwendung (66) eine authentische digitale Signatur und eine Signaturidentifikation umfasst, wenn die eine oder mehrere APIs, zu der/denen die Softwareanwendung (66) einen Zugang erfordert, eine sensitive API umfasst/umfassen und die Softwareanwendung (66) eine authentische globale Signatur umfasst; und

> der Schritt des Verweigerns eines Zugangs zu der einen oder mehreren APIs für die Softwareanwendung (66) die Schritte aufweist:

>> Verweigern eines Zugangs zu der einen oder mehreren APIs für die Softwareanwendung (66), wenn die Softwareanwendung (66) keine authentische globale Signatur umfasst; und
>> Verweigern eines Zugangs zu der sensitiven API für die Softwareanwendung (66), wenn die

eine oder mehrere APIs, zu der/denen die Softwareanwendung (66) einen Zugang erfordert, eine sensitive API umfasst/umfassen, die Softwareanwendung (66) eine authentische globale Signatur umfasst,

und die Softwareanwendung (66) keine authentische digitale Signatur und Signaturidentifizierer umfasst, die erforderlich sind für einen Zugang zu der sensitiven API.

Revendications

1. Procédé destiné à limiter l'accès à des interfaces de programmation applicative sur un dispositif mobile (62) comprenant les étapes qui consistent à :

    charger une application logicielle (66) ayant une signature numérique (96) et une identification de signature (94) sur le dispositif mobile (62) qui nécessite l'accès à une ou plusieurs interfaces de programmation applicative (APIs) ayant au moins un identifiant de signature (92),
    authentifier la signature numérique (96) lorsque l'identification de signature (94) correspond à l'identifiant de signature (92), et
    refuser à l'application logicielle (66) l'accès à l'une ou aux plusieurs APIs lorsque l'application logicielle (66) ne comporte pas de signature numérique authentique (96).

2. Procédé de la revendication 1, dans lequel la signature numérique (96) et l'identification de signature (94) sont associées à un type de dispositif mobile (62).

3. Procédé de la revendication 1 ou 2, comprenant l'étape supplémentaire qui consiste à :

    éliminer l'application logicielle (66) du dispositif mobile (62) lorsque l'application logicielle (66) ne comporte pas de signature numérique authentique (96).

4. Procédé d'une des revendications précédentes, dans lequel :

    l'application logicielle (66) comporte une pluralité de signatures numériques (96) et d'identifications de signatures (94) ; et
    la pluralité de signatures numériques (96) et d'identifications de signatures (94) comporte des signatures numériques et des identifications de signatures respectivement associées à différents types de dispositifs mobiles (62).

5. Procédé de la revendication 4, dans lequel chacune de la pluralité de signatures numériques (96) et des

identifications de signatures associées (94) est générée par une autorité de signature de code correspondante respective.

6. Procédé d'une des revendications précédentes, dans lequel l'étape qui consiste à déterminer si l'application logicielle (66) comporte une signature numérique authentique (96) comprend les étapes supplémentaires qui consistent à

    vérifier l'authenticité de la signature numérique (96) lorsqu'il y a correspondance entre l'identification de signature (94) et des identifiants respectifs de l'au moins un identifiant de signature (92).

7. Procédé de la revendication 5, dans lequel chacune de la pluralité de signatures numériques (96) et des identifications de signatures (94) sont générées par son autorité de signature de code correspondante en appliquant une clé de signature privée respective associée à l'autorité de signature de code à une empreinte numérique de l'application logicielle (66).

8. Procédé d'une des revendications précédentes, dans lequel l'étape qui consiste à authentifier la signature numérique (96) lorsqu'il y a correspondance entre l'identification de signature (94) et l'identifiant de signature (92) comprend les étapes qui consistent à :

    vérifier que l'identification de signature (94) correspond à l'identifiant de signature (92) authentifiant la signature numérique (96) où l'identification de signature (94) correspond à l'identifiant de signature (92) comprenant les étapes qui consistent à :

        stocker une clé de signature publique sur un dispositif mobile (62) qui correspond à la clé de signature privée associée à l'autorité de signature de code qui génère la signature numérique (96) ;
        générer une empreinte numérique de l'application logicielle (66) pour obtenir une empreinte numérique générée ;
        appliquer la clé de signature publique à la signature numérique (96) afin d'obtenir une empreinte numérique récupérée ; et
        comparer l'empreinte numérique générée à l'empreinte numérique récupérée.

9. Procédé d'une des revendications précédentes, dans lequel :

    le dispositif mobile (62) comporte une pluralité d'APIs;
    au moins l'une de la pluralité d'APIs est classi-

Page 596 of 1415

fiée comme sensible ;
l'accès à l'une de la pluralité d'APIs nécessite
une signature globale authentique ;
l'accès à chacune de la pluralité d'APIs sensi-
bles nécessite une signature globale authenti-
que et une signature numérique authentique as-
sociée à une identification de signature (94) ;
l'étape qui consiste à déterminer si l'application
logicielle (66) comporte une signature numéri-
que authentique et une identification de signa-
ture comprend les étapes qui consistent à :

> déterminer si l'une ou les plusieurs APIs
> auxquelles l'application logicielle (66) né-
> cessite un accès comporte une API
> sensible ;
> déterminer si l'application logicielle (66)
> comporte une signature globale
> authentique ; et
> déterminer si l'application logicielle (66)
> comporte une signature numérique authen-
> tique et une identification de signature où
> l'une ou plusieurs APIs auxquelles l'appli-
> cation logicielle (66) nécessite un accès
> comporte une API sensible et l'application
> logicielle (66) comporte une signature glo-
> bale authentique ; et
> l'étape qui consiste à refuser l'accès de l'ap-
> plication logicielle (66) à l'une ou à plusieurs
> APIs comprend les étapes qui consistent à :

>> refuser l'accès de l'application logiciel-
>> le (66) à l'une ou aux plusieurs APIs
>> lorsque l'application logicielle (66) ne
>> comporte pas de signature globale
>> authentique ; et
>> refuser l'accès de l'application logiciel-
>> le (66) à l'API sensible lorsque une ou
>> plusieurs APIs auxquelles l'application
>> logicielle (66) nécessite un accès com-
>> porte une API sensible, l'application lo-
>> gicielle (66) comporte une signature
>> globale authentique, et l'application lo-
>> gicielle (66) ne comporte pas de signa-
>> ture numérique authentique et d'identi-
>> fiant de signature nécessaires pour
>> l'accès à l'API sensible.
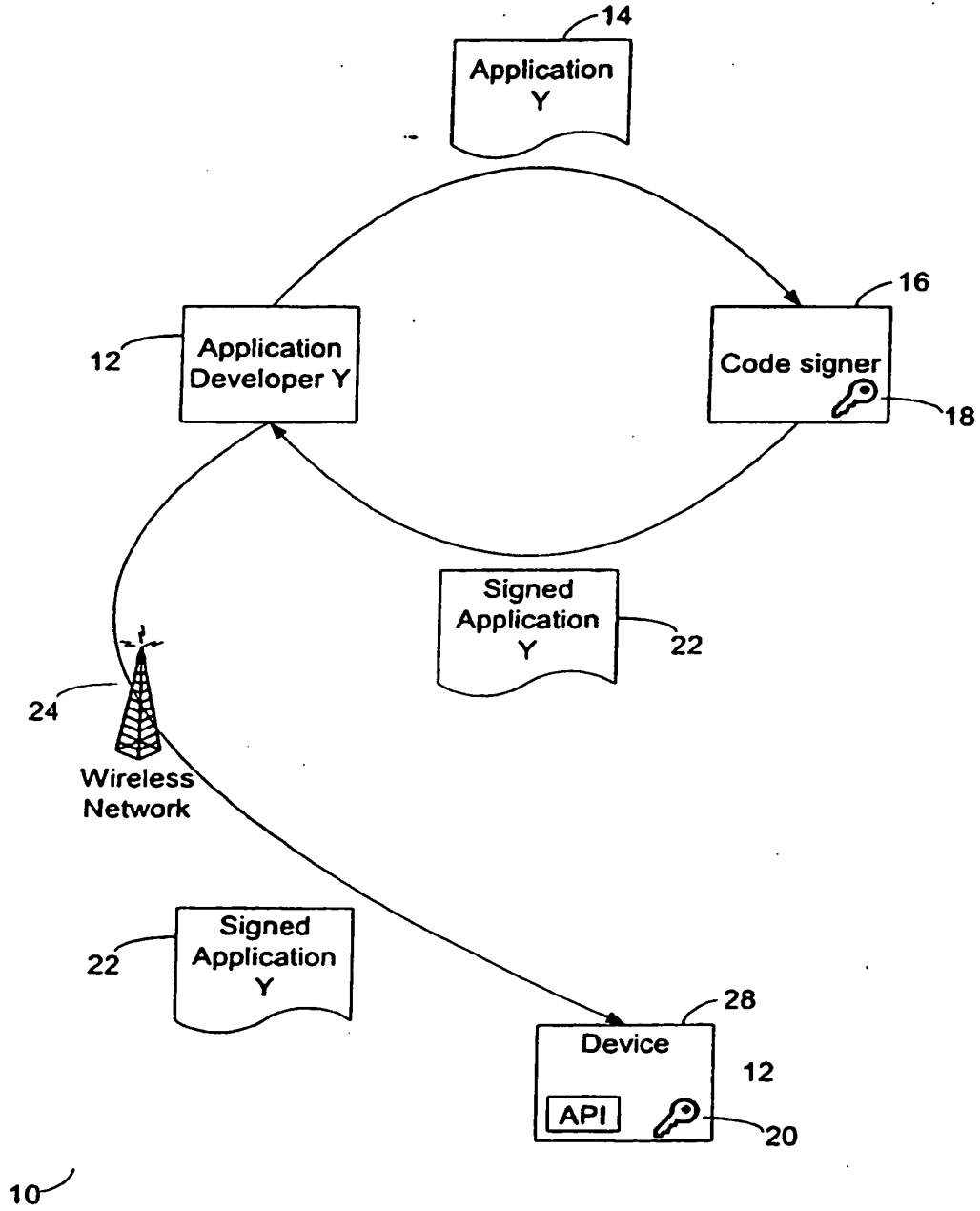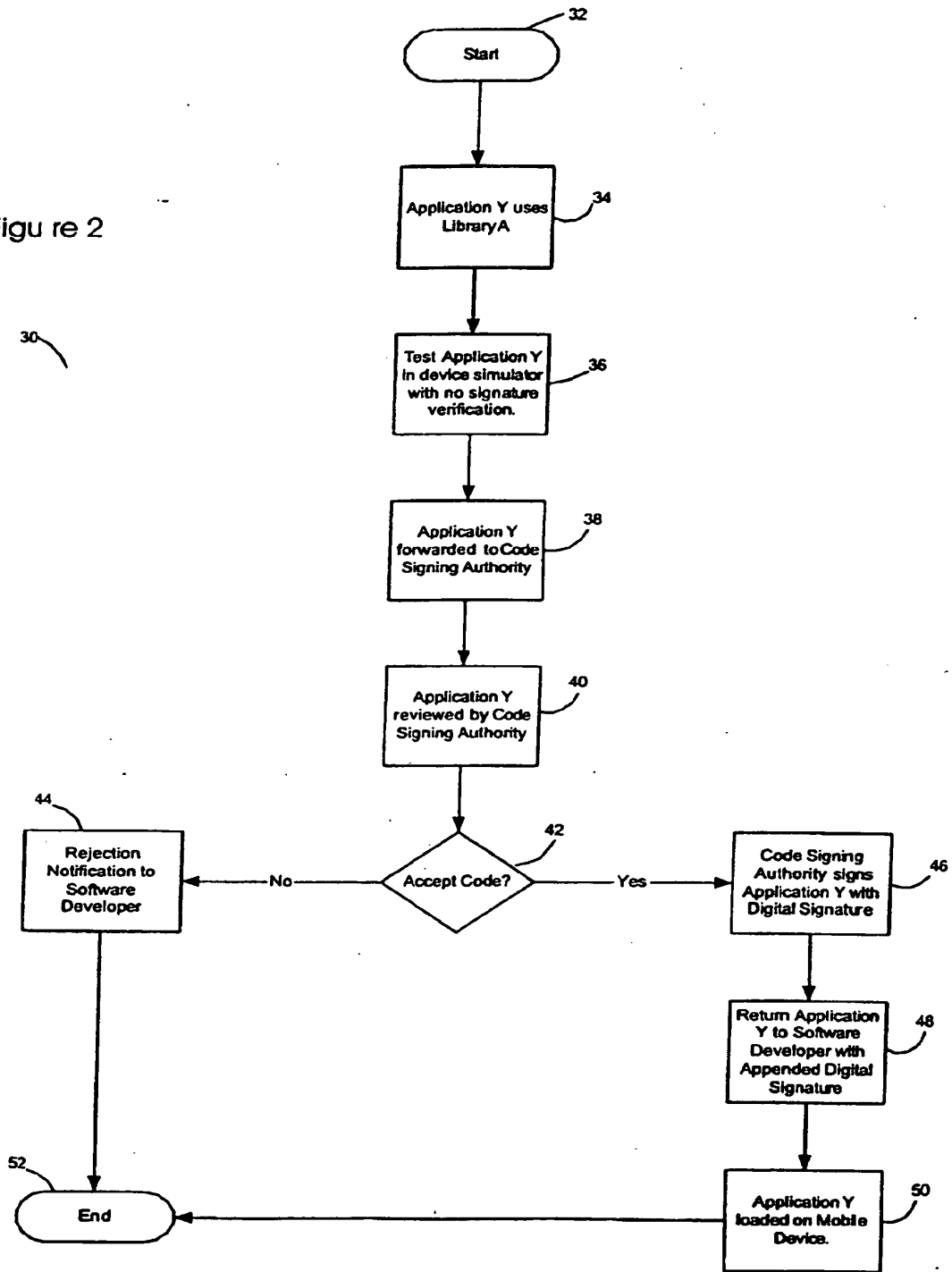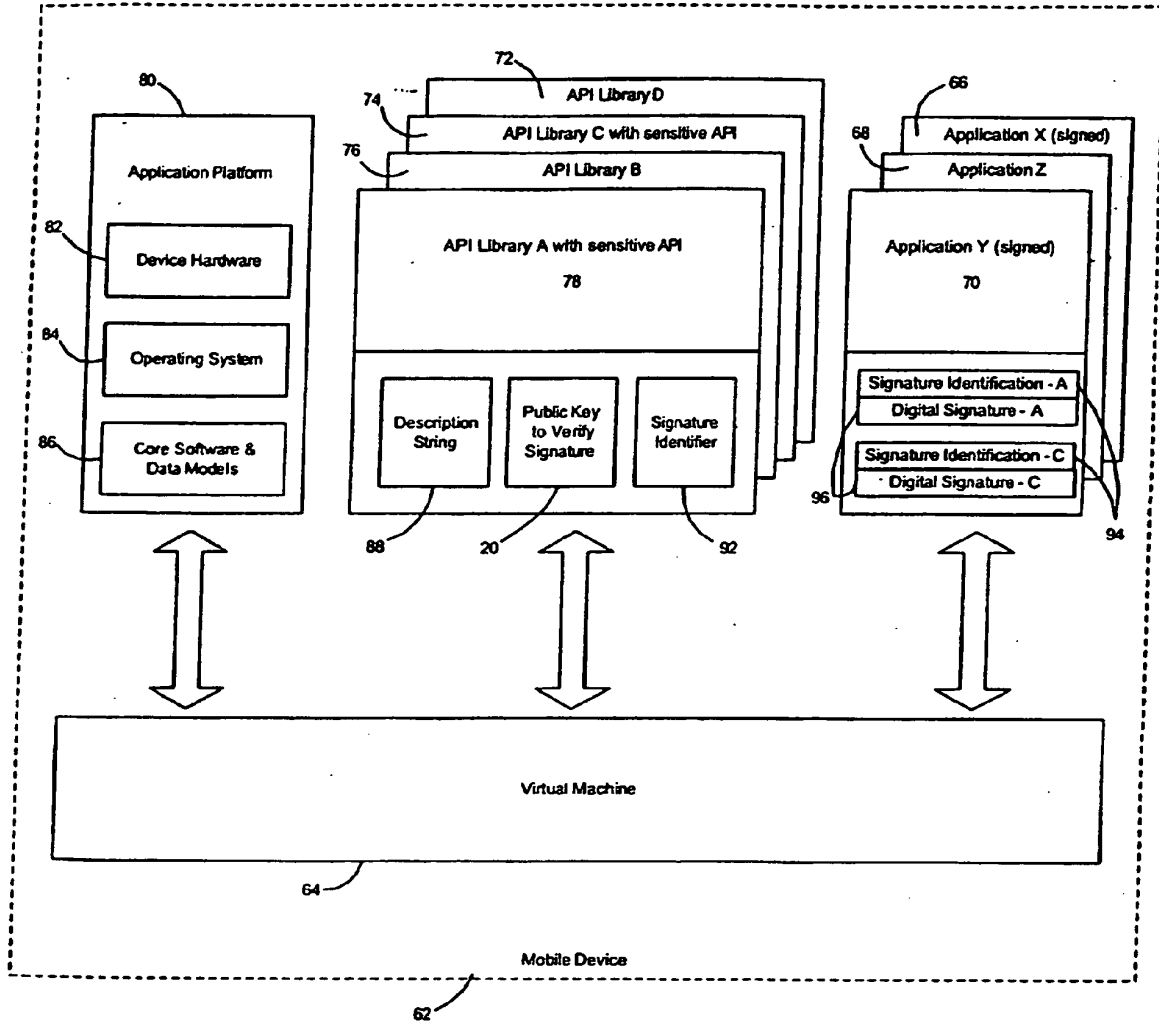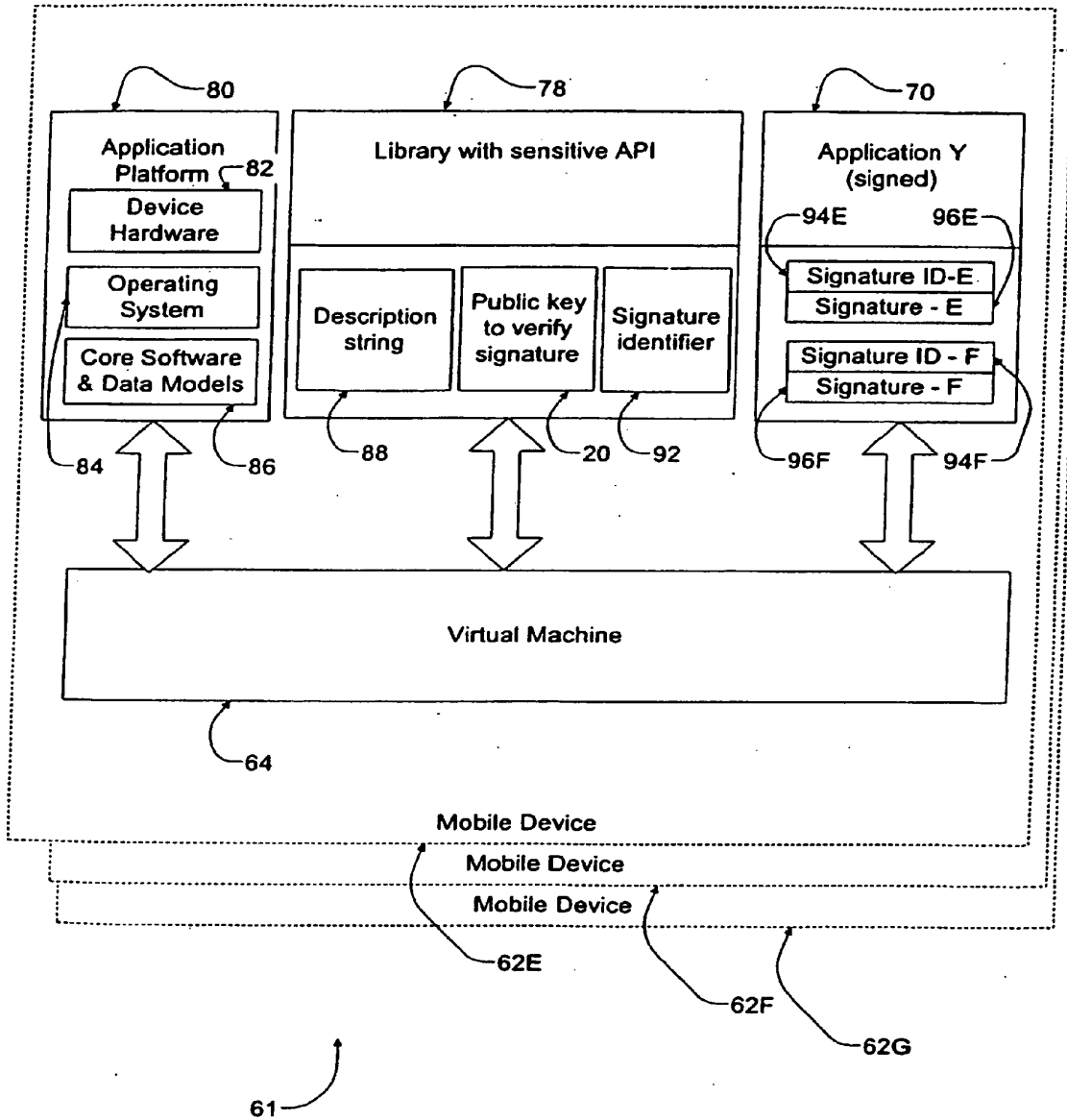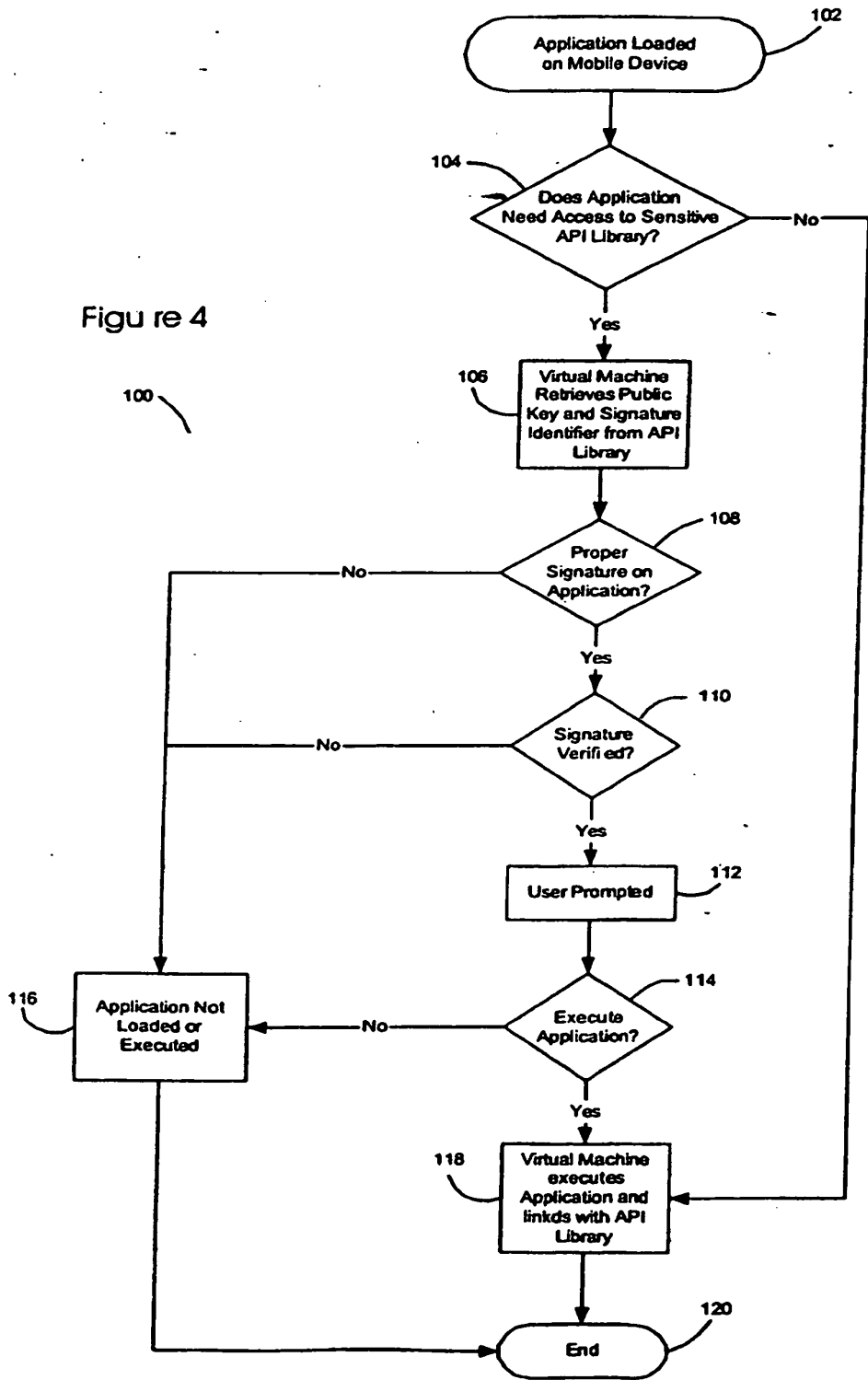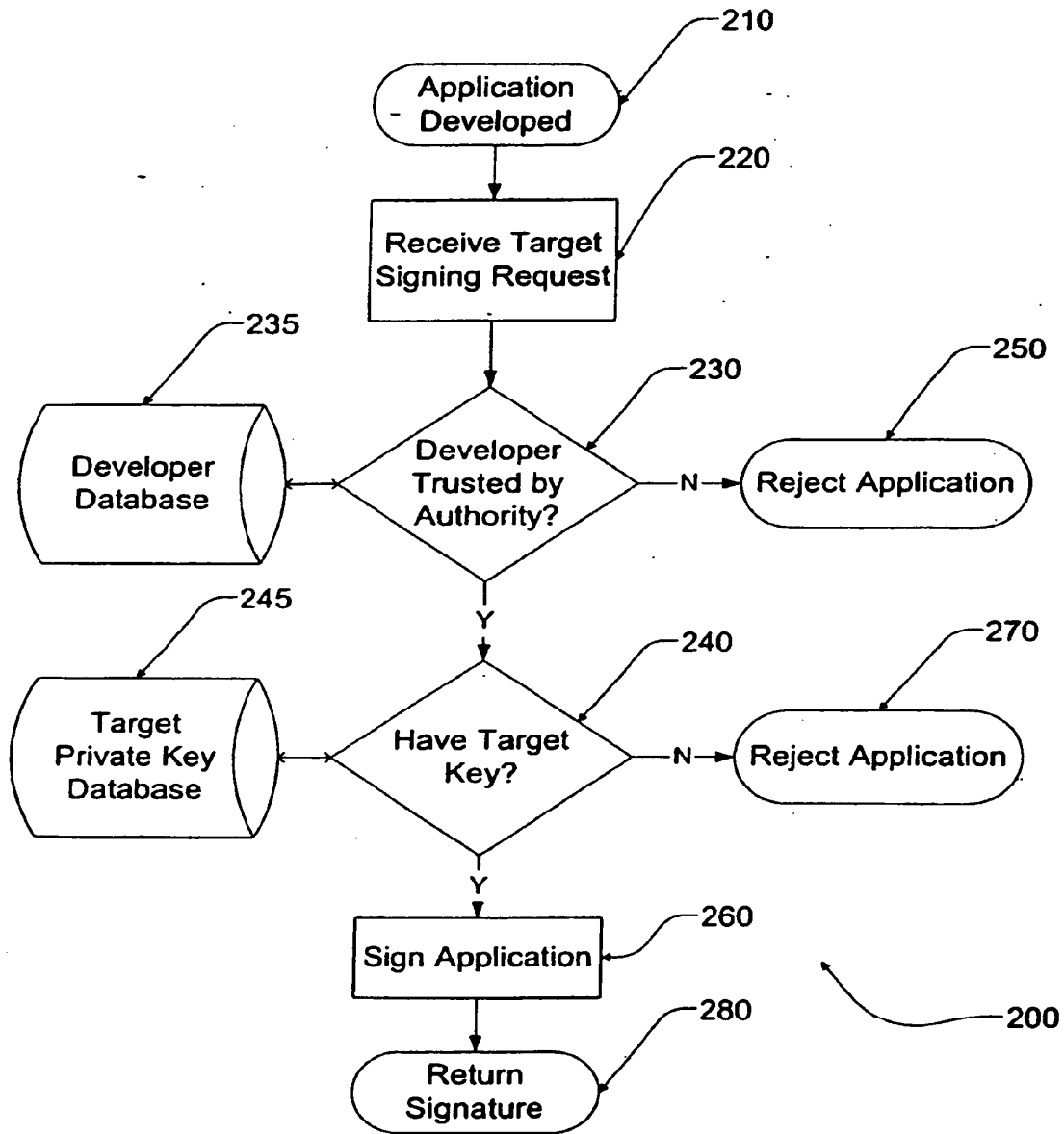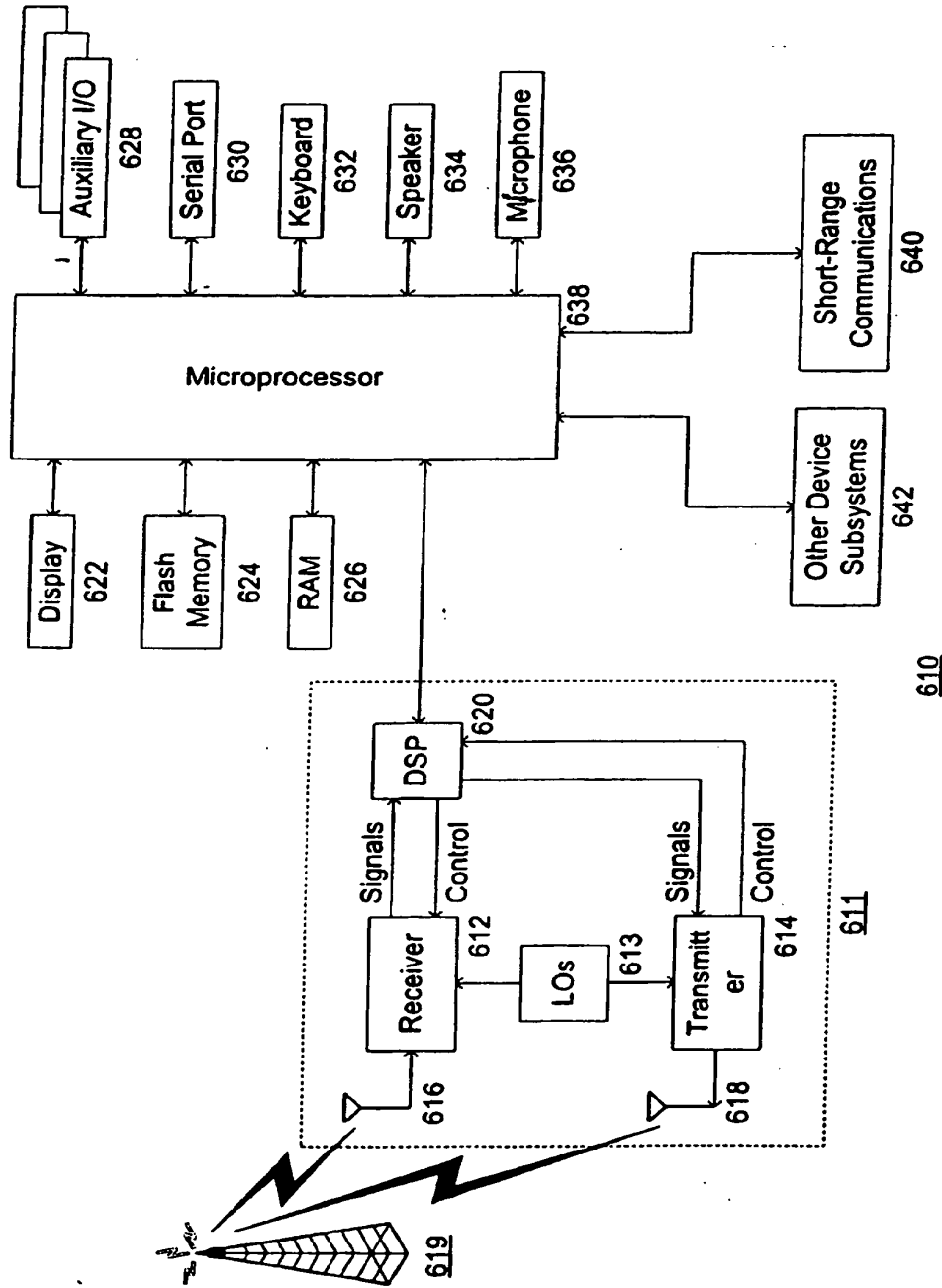
Figure 1

Figu re 2

Figure 3

Figure 3A

20

Figure 4

Figure 5

Figure 6

610

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- Handbuch der Chipkarten. 1999, vol. 3 **[0003]**

香港特別行政區政府知識產權署專利註冊處
Patents Registry, Intellectual Property Department
The Government of the Hong Kong Special Administrative Region

專利註冊紀錄冊 REGISTER OF PATENTS
註冊紀錄冊記項 REGISTER ENTRY

申請編號 **Application No.**	:06108036.3
提交日期 **Filing date**	:18.07.2006
法律程序所用語文 **Language of Proceedings**	:En
聲稱享有的優先權 **Priority claimed**	:21.09.2000 US 234152P :26.09.2000 US 235354P :20.02.2001 US 270663P
發表編號 **Publication No.**	:HK1091666
專利說明書首次發表日期 **Date of first publication**	:26.01.2007/A
歐洲專利發表編號 **EP Publication No.**	:EP 1626324
歐洲專利申請發表日期 **EP Application Publication Date**	:15.02.2006
歐洲專利申請編號 **EP Application No.**	:05024661.0
歐洲專利申請提交日期 **EP Application Filing Date**	:20.09.2001

發明名稱 **Title**
軟件編碼標記系統和方法
SOFTWARE CODE SIGNING SYSTEM AND METHOD

申請人 **Applicant**
RESEARCH IN MOTION LIMITED
295 Phillip Street
Waterloo
Ontario N2L 3W8
CANADA

發明人 **Inventor**
Yach, David P.

Brown, Michael S.

Little, Herbert A.

分類 **Classified to**	:G06F

送達地址 **Address for Service**
Williams, Davis, Hill & Co
Suite 701, 7th Floor
6-8 Pottinger Street, Central
Hong Kong

代理人地址 **Agent's Address**
Suite 701, 7th Floor, 6-8 Pottinger Street, Central, Hong Kong

狀況 Status	申請有效 Application in force

18.03.2010   更改送達地址 Change of Address for Service

Change of address for service to Vivien Chan & Co., 38/F, Cosco
Tower, Grand Millennium Plaza, 183 Queen's Road Central, Hong
Kong.
送達地址已更改為 陳韻雲律師行 香港上環皇后大道中183號新紀元廣場中遠
大廈38樓。

註冊紀錄冊記項完結
**** END OF REGISTER ENTRY ****

維持標準專利申請詳情
MAINTENANCE DETAILS OF STANDARD PATENT APPLICATION

發表編號 **PUBLICATION NO.**   :HK1091666

申請人 **APPLICANT**
RESEARCH IN MOTION LIMITED
295 Phillip Street
Waterloo
Ontario N2L 3W8
CANADA

指定專利申請提交日期   :20.09.2001
DATE OF FILING OF DESIGNATED PATENT APPLICATION

記錄請求發表日期   :26.01.2007
DATE OF PUBLICATION OF REQUEST TO RECORD

維持費到期繳交日期   :20.09.2012
DATE OF MAINTENANCE FEE DUE

上次維持費繳交日期 **DATE OF LAST MAINTENANCE**   :

提交註冊及批予專利申請日期   :
APPLICATION FOR REGISTRATION & GRANT FILING DATE

狀況   申請有效
STATUS   Application in force

報告完結
**** END OF REPORT ****

[12]

STANDARD PATENT SPECIFICATION
標準專利說明書

[54] SOFTWARE CODE SIGNING SYSTEM AND METHOD 軟件編碼標記系統和方法

[73] Proprietor 專利所有人

RESEARCH IN MOTION LIMITED

295 Phillip Street, Waterloo

Ontario N2L 3W8

CANADA

[72] Inventor 發明人

YACH, DAVID P.

BROWN, MICHAEL S.

LITTLE, HERBERT A.

[74] Agent and / or address for service 代理人及/或送達地址

VIVIEN CHAN & CO.

38/F Cosco Tower, Grand Millennium Plaza

183 Queen's Road Central, HONG KONG

[12]
STANDARD PATENT SPECIFICATION
標準專利說明書

[54] SOFTWARE CODE SIGNING SYSTEM AND METHOD 軟件編碼標記系統和方法

[73] Proprietor 專利所有人

RESEARCH IN MOTION LIMITED

295 Phillip Street, Waterloo

Ontario N2L 3W8

CANADA

[72] Inventor 發明人

YACH, DAVID P.

BROWN, MICHAEL S.

LITTLE, HERBERT A.

[74] Agent and / or address for service 代理人及/或送達地址

VIVIEN CHAN & CO.

38/F Cosco Tower, Grand Millennium Plaza

183 Queen's Road Central, HONG KONG

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 1/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 01819200.9

[45] 授权公告日 2009 年 12 月 23 日

[11] 授权公告号 CN 100573402C

[54] 发明名称
　　代码签字系统及方法

[57] 摘要
　　提供了一种代码签字系统和方法。代码签字系统与有数字签字的软件应用程序一起工作，并包括应用平台、应用程序编程接口（API）和虚拟机。API 用来把软件应用程序与应用平台相链接。虚拟机验证数字签字的真实性，以控制软件应用程序访问 API。
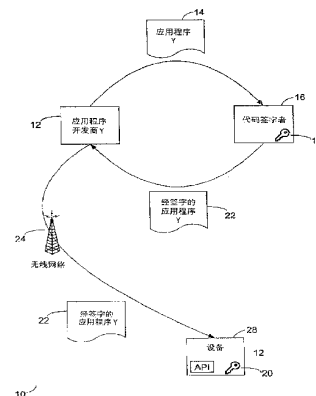
1．一种代码签字系统，用于与具有数字签字和签字标识的软件应用程序一起工作，其中，数字签字与签字标识相关，包括：

应用平台；

应用编程接口 API，具有关联的签字标识符，设置 API 将软件应用程序和应用平台链接；

虚拟机，如果签字标识符对应签字标识，则为了控制软件应用程序访问 API，虚拟机验证数字签字的真实性，

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字，所述虚拟机通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用签字密钥应用到数字签字中来获得恢复的杂乱信号、比较产生的杂乱信号和恢复的杂乱信号来验证数字签字的真实性。

2．根据权利要求 1 所述的代码签字系统，其特征在于如果数字签字不真实，则虚拟机拒绝软件应用程序访问 API。

3．根据权利要求 1 所述的代码签字系统，其特征在于如果数字签字不真实，则虚拟机删除软件应用程序。

4．根据权利要求 1 所述的代码签字系统，其特征在于代码签字系统装在移动设备上。

5．根据权利要求 1 所述的代码签字系统，其特征在于数字签字由代码签字授权机构产生。

6．根据权利要求 1 所述的代码签字系统，其特征在于还包括：

多个 API 程序库，每个 API 程序库包括多个 API，其中，虚拟机通过软件应用程序控制访问多个 API 程序库。

7．根据权利要求 6 所述的代码签字系统，其特征在于：

至少一个 API 程序库被分类为敏感的；

访问敏感的 API 程序库要求将数字签字与签字标识关联，其中，签字标识对应与敏感的 API 程序库关联的签字标识符；

软件应用程序包括至少一个数字签字和至少一个关联的签字标识，用

2

于访问敏感的 API 程序库；

虚拟机通过验证包括在软件应用程序中的一个数字签字来授权软件应用程序访问敏感的 API 程序库，所述软件应用程序具有对应敏感的 API 程序库的签字标识符的签字标识。

8．根据权利要求 7 所述的代码签字系统，其特征在于敏感的 API 程序库还包括描述字符串，其中，当软件应用程序试图访问敏感的 API 时，显示描述字符串。

9．根据权利要求 1 所述的代码签字系统，其特征在于应用平台包括操作系统。

10．根据权利要求 1 所述的代码签字系统，其特征在于包括一个或多个移动设备的核心功能。

11．根据权利要求 1 所述的代码签字系统，其特征在于包括移动设备上的硬件。

12．根据权利要求 11 所述的代码签字系统，其特征在于硬件包括用户身份模块卡。

13．根据权利要求 1 所述的代码签字系统，其特征在于软件应用程序是用于移动设备的 Java 应用程序。

14．根据权利要求 1 所述的代码签字系统，其特征在于 API 与应用平台上的加密流程接口。

15．根据权利要求 1 所述的代码签字系统，其特征在于 API 与应用平台上的专用数据模块接口。

16．根据权利要求 1 所述的代码签字系统，其特征在于虚拟机是安装在移动设备上的 Java 虚拟机。

17．一种控制在移动设备上访问敏感的应用程序编程接口的方法，包括下列步骤：

把软件应用程序装到移动设备上，所述软件应用程序要求访问具有签字标识符的敏感的应用程序编程接口 API；

确定软件应用程序是否包括数字签字和签字标识；

如果签字标识不与签字标识符对应，那么拒绝软件应用程序访问敏感的 API；

3

如果签字标识与签字标识符对应，那么验证数字签字的真实性，其中，基于数字签字的真实性的验证，由软件应用程序访问敏感的API，

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字，其中，验证数字签字的真实性包括步骤：

在移动设备上存储对应专用签字密钥的公用签字密钥；

产生软件应用程序的杂乱信号来获得产生的杂乱信号；

将公用签字密钥应用到数字签字中来获得恢复的杂乱信号；

比较产生的杂乱信号和恢复的杂乱信号。

18．根据权利要求17所述的方法，其特征在于还包括步骤：如果签字标识不对应签字标识符，则从移动设备删除软件应用程序。

19．根据权利要求17所述的方法，其特征在于数字签字和签字标识由代码签字授权机构产生。

20．根据权利要求17所述的方法，其特征在于还包括步骤：

如果数字签字不真实，则拒绝软件应用程序访问敏感的API。

21．根据权利要求17所述的方法，其特征在于还包括步骤：如果数字签字不真实，则从移动设备上删除软件应用程序。

22．根据权利要求17所述的方法，其特征在于当软件应用程序试图访问所述的敏感的API时，向用户显示描述字符串。

23．根据权利要求17所述的方法，其特征在于还包括如下步骤：

显示描述字符串，所述描述字符串通知移动设备的用户软件应用程序要求访问敏感的API。

24．根据权利要求17所述的方法，其特征在于还包括步骤：

从用户接收指令，准许或拒绝软件应用程序访问敏感的API。

25．一种移动设备，包括：

应用平台，具有应用编程接口API；

虚拟机，用于验证由各个软件应用程序提供的数字签字和签字标识，以便访问API；

在软件应用程序提供的数字签字由代码签字协议验证后，虚拟机也允许软件应用程序访问至少一个API；

代码签字授权机构向要求访问至少一个API的软件应用程序提供数

4

字签字和签字标识，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字，其中，提供给软件应用程序的签字标识包括仅被授权的签字标识，以便允许访问多个移动设备的第一子设备；

其中，第一数字签字和第一签字标识用于第一种类型的移动设备；

第二数字签字和第二签字标识用于第二种类型的移动设备；

与应用程序关联的第一数字签字和第一签字标识防止使用第二种类型移动设备上的 API 的应用程序；

与应用程序关联的第二数字签字和第二签字标识防止使用第一种类型移动设备上的 API 的应用程序，

其中，虚拟机通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用签字密钥应用到数字签字中来获得恢复的杂乱信号、比较产生的杂乱信号和恢复的杂乱信号来验证第一数字签字或第二数字签字的真实性。

26．根据权利要求 25 所述的移动设备，其特征在于虚拟机包括验证系统和控制系统，其中，虚拟机是 Java 虚拟机，软件应用程序是 Java 应用程序。

27．根据权利要求 25 所述的移动设备，其特征在于控制系统为至少一个 API 的每个程序库要求一个数字签字和一个签字标识。

28．根据权利要求 25 所述的移动设备，其特征在于应用平台的 API 至少接入执行加密算法的加密模块、数据存储器、专用数据模型和用户接口之一。

29．根据权利要求 25 所述的移动设备，其特征在于至少一个 API 被分类为敏感的，敏感的 API 还包括描述字符串，其中，当软件应用程序试图访问敏感的 API 时，描述字符串被显示给用户。

30．根据权利要求 25 所述的移动系统，其特征在于第一种类型的移动设备和第二种类型的移动设备是不同类型的移动设备。

31．一种在移动设备上控制软件开发商开发的软件应用程序访问具有签字标识符的应用程序编程接口 API 的方法，包括如下步骤：

从软件开发商接收软件应用程序；

确定软件应用程序是否满足至少一个标准；

5

如果软件应用程序满足至少一个标准，则把数字签字和签字标识添加到软件应用程序；

如果签字标识对应签字标识符，则验证添加到软件应用程序的数字签字的真实性；

如果数字签字是真实的，向软件应用程序提供到 API 的路径；

把数字签字和签字标识添加到软件应用程序的步骤包括产生数字签字，包括下列步骤：

计算软件应用程序的杂乱信号；

把专用签字密钥应用到软件应用程序的杂乱信号，以产生数字签字；

在移动设备上提供公用签字密钥；

在移动设备上计算软件应用程序的杂乱信号以获得计算的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

通过比较计算的杂乱信号与恢复的杂乱信号来验证数字签字。

32．根据权利要求 31 所述的方法，其特征在于确定软件应用程序是否满足至少一个标准的步骤由代码签字授权机构执行。

33．根据权利要求 32 所述的方法，其特征在于使用安全的杂乱信号算法计算软件应用程序的杂乱信号。

34．根据权利要求 31 所述的方法，其特征在于进一步包括，如果数字签字不真实，则拒绝该软件应用程序访问 API。

35．一种在移动设备上控制访问具有签字标识符的敏感应用编程接口 API 的方法，包括步骤：

注册一个或多个可信的软件开发商，编制访问敏感的 API 的软件应用程序；

接收软件应用程序的杂乱信号；

确定杂乱信号是否是注册的软件开发商所发送；

产生数字签字，其中，

数字签字和签字标识被添加到软件应用程序；

如果签字标识对应签字标识符，为了控制软件应用程序访问敏感的 API，移动设备验证数字签字的真实性；

产生数字签字的步骤是把专用签字密钥应用到软件应用程序的杂乱

6

信号执行的，所述杂乱信号由注册的软件开发商所发送；

其中，移动设备执行下列附加的步骤验证数字签字的真实性：

在移动设备上提供公用签字密钥；

在移动设备上计算软件应用程序的杂乱信号，以获得计算的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

通过比较计算的杂乱信号与恢复的杂乱信号，以确定数字签字是否真实；

如果数字签字不真实，则拒绝软件应用程序访问敏感的 API。

36．根据权利要求 35 所述的方法，其特征在于产生数字签字的步骤由代码签字授权机构执行。

37．一种在移动设备上限制访问应用编程接口的方法，包括如下步骤：

把具有数字签字和签字标识的软件应用程序装到要求访问一个或多个具有至少一个签字标识符的 API 的移动设备上；

如果签字标识对应签字标识符，则验证数字签字；

如果软件应用程序不包括真实的数字签字，则拒绝软件应用程序访问一个或多个 API；

其中，如果签字标识与签字标识符对应，则验证数字签字的步骤包括：

验证与签字标识符对应的签字标识；

把公用签字密钥存储到移动设备上，该公用签字密钥对应与代码签字授权机构关联的专用签字密钥，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字；

产生软件应用程序的杂乱信号，以获得产生的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

将产生的杂乱信号与恢复的杂乱信号进行比较。

38．根据权利要求 37 所述的方法，其特征在于数字签字和签字标识与移动设备的类型有关。

39．根据权利要求 37 所述的方法，其特征在于包括附加的步骤：

如果软件应用程序不包括真实的数字签字，则从移动设备上消除该软件应用程序。

7

40．根据权利要求 37 所述的方法，其特征在于：

软件应用程序包括多个数字签字和签字标识；

多个数字签字和签字标识分别包括与各不同类型的移动设备有关的数字签字和签字标识。

41．根据权利要求 40 所述的方法，其特征在于每个数字签字和有关的签字标识是由各相应的代码签字授权机构产生的。

42．根据权利要求 40 所述的方法，其特征在于通过把与代码签字授权机构有关的各个专用签字密钥应用到软件应用程序的杂乱信号，由对应的代码签字授权机构产生每个数字签字和签字标识。

43．一种控制软件应用程序访问具有签字标识符的应用编程接口 API 的方法，软件应用程序具有数字签字和签字标识，包括：

如果签字标识对应于签字标识符，则验证数字签字的真实性；

如果软件应用程序提供的数字签字是真实的，允许访问至少一个 API；

软件应用程序的数字签字和签字标识由代码签字授权机构产生；

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字；

通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用签字密钥应用到数字签字中来获得恢复的杂乱信号、验证产生的杂乱信号和恢复的杂乱信号是否相同来验证数字签字。

44．根据权利要求 43 所述的方法，其特征在于如果软件应用程序提供的数字签字被验证，则允许访问 API 的程序库。

45．根据权利要求 43 所述的方法，其特征在于 API 至少接入执行加密算法的加密模块、数据存储器、专用数据模型和用户接口之一。

46．根据权利要求 43 所述的方法，其特征在于至少一个 API 被分类为敏感的，敏感的 API 还包括描述字符串，其中，当软件应用程序试图访问敏感的 API 时，向用户显示描述字符串。

47．根据权利要求 43 所述的方法，其特征在于 API 提供访问至少一个或多个移动设备的核心功能、操作系统和移动设备上的硬件。

48．根据权利要求 43 所述的方法，其特征在于要求软件应用程序提

8

供全局数字签字的验证，以访问任何 API。

9

代码签字系统及方法

有关申请的参照

　　本申请要求下列申请的优先权：

　　"代码签字系统及方法"于2000年9月21申请的美国临时申请，申请号是60/234152；"代码签字系统及方法"于2000年9月22申请的美国临时申请，申请号是60/235354；"代码签字系统及方法"于2001年2月20申请的美国临时申请，申请号是60/270663；

技术领域

　　本发明涉及软件应用程序的安全协议领域。更具体地说，本发明提供代码签字系统及方法，特别适用于移动通信设备的 Java^{TM} 应用程序，例如个人数字助理、蜂窝电话，无线双程通信设备（以下通称为"移动设备"或简称"设备"）。

背景技术

　　包括软件代码签字方案的安全协议是众所周知的，典型地，这种安全协议用来保证从互联网下载的软件应用程序的可靠性。在典型的代码签字方案中，数字签字附于识别软件开发商的软件应用程序。一旦该软件被用户下载，用户必须只根据对软件开发商信誉的了解来判断该软件应用程序的可靠性。这类代码签字方案不能保证由第三方为移动设备所写的软件应用程序适合与本地应用程序和其它资源交互作用。因为典型的代码签字协议是不安全的，且只依赖于用户的判断，有严重破坏的风险，"特洛伊木马"型软件应用程序可能被下载并安装在移动设备上。

　　网络工作者还需要一种系统和方法，来控制软件应用程序在移动设备上起动。

　　还进一步需要 2.5G 和 3G 网络，其中合作客户或网络工作者都喜欢控制在设备上发布给其顾员的软件类型。

10

发明内容

本发明的目的是提供代码签字系统和方法。

按照本发明的一方面，一种代码签字系统，用于与具有数字签字和签字标识的软件应用程序一起工作，其中，数字签字与签字标识相关，包括：

应用平台；

应用编程接口 API，具有关联的签字标识符，设置 API 将软件应用程序和应用平台链接；

虚拟机，如果签字标识符对应签字标识，则为了控制软件应用程序访问 API，虚拟机验证数字签字的真实性，

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字，所述虚拟机通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用数字签字密钥应用到数字签字中来获得恢复的杂乱信号、比较产生的杂乱信号和恢复的杂乱信号来验证数字签字的真实性。

按照本发明的另一方面，一种控制在移动设备上访问敏感的应用程序编程接口的方法，包括下列步骤：

把软件应用程序装到移动设备上，所述软件应用程序要求访问具有签字标识符的敏感的应用程序编程接口 API；

确定软件应用程序是否包括数字签字和签字标识；

如果签字标识不与签字标识符对应，那么拒绝软件应用程序访问敏感的 API；

如果签字标识与签字标识符对应，那么验证数字签字的真实性，其中，基于数字签字的真实性的验证，由软件应用程序访问敏感的 API，

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字，其中，验证数字签字的真实性包括步骤：

在移动设备上存储对应专用签字密钥的公用签字密钥；

产生软件应用程序的杂乱信号来获得产生的杂乱信号；

将公用签字密钥应用到数字签字中来获得恢复的杂乱信号；

比较产生的杂乱信号和恢复的杂乱信号。

按照本发明的另一方面，一种移动设备，包括：

11

应用平台，具有应用编程接口 API；

虚拟机，用于验证由各个软件应用程序提供的数字签字和签字标识，以便访问 API；

在软件应用程序提供的数字签字由代码签字协议验证后，虚拟机也允许软件应用程序访问至少一个 API；

代码签字授权机构向要求访问至少一个 API 的软件应用程序提供数字签字和签字标识，根据签字标识的签字方案和使用软件应用程序的杂乱信号产生用于软件应用程序的数字签字，其中，提供给软件应用程序的签字标识包括仅被授权的签字标识，以便允许访问多个移动设备的第一子设备；

其中，第一数字签字和第一签字标识用于第一种类型的移动设备；

第二数字签字和第二签字标识用于第二种类型的移动设备；

与应用程序关联的第一数字签字和第一签字标识防止使用第二种类型移动设备上的 API 的应用程序；

与应用程序关联的第二数字签字和第二签字标识防止使用第一种类型移动设备上的 API 的应用程序，

其中，虚拟机通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用数字签字密钥应用到数字签字中来获得恢复的杂乱信号、比较产生的杂乱信号和恢复的杂乱信号来验证第一数字签字或第二数字签字的真实性。

按照本发明的另一方面，一种在移动设备上控制软件开发商开发的软件应用程序访问具有签字标识符的应用程序编程接口 API 的方法，包括如下步骤：

从软件开发商接收软件应用程序；

确定软件应用程序是否满足至少一个标准；

如果软件应用程序满足至少一个标准，则把数字签字和签字标识添加到软件应用程序；

如果签字标识对应签字标识符，则验证添加到软件应用程序的数字签字的真实性；

如果数字签字是真实的，向软件应用程序提供到 API 的路径；

12

把数字签字和签字标识添加到软件应用程序的步骤包括产生数字签字，包括下列步骤：

计算软件应用程序的杂乱信号；

把专用签字密钥应用到软件应用程序的杂乱信号，以产生数字签字；

在移动设备上提供公用签字密钥；

在移动设备上计算软件应用程序的杂乱信号以获得计算的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

通过比较计算的杂乱信号与恢复的杂乱信号来验证数字签字。

按照本发明的另一方面，一种在移动设备上控制访问具有签字标识符的敏感应用编程接口 API 的方法，包括步骤：

注册一个或多个可信的软件开发商,编制访问敏感的 API 的软件应用程序；

接收软件应用程序的杂乱信号；

确定杂乱信号是否是注册的软件开发商所发送；

产生数字签字，其中，

数字签字和签字标识被添加到软件应用程序；

如果签字标识对应签字标识符，为了控制软件应用程序访问敏感的 API，移动设备验证数字签字的真实性；

产生数字签字的步骤是把专用签字密钥应用到软件应用程序的杂乱信号执行的，所述杂乱信号由注册的软件开发商所发送；

其中，移动设备执行下列附加的步骤验证数字签字的真实性：

在移动设备上提供公用签字密钥；

在移动设备上计算软件应用程序的杂乱信号，以获得计算的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

通过比较计算的杂乱信号与恢复的杂乱信号，以确定数字签字是否真实；

如果数字签字不真实，则拒绝软件应用程序访问敏感的 API。

按照本发明的另一方面，一种在移动设备上限制访问应用编程接口的方法，包括如下步骤：

13

把具有数字签字和签字标识的软件应用程序装到要求访问一个或多个具有至少一个签字标识符的 API 的移动设备上；

如果签字标识对应签字标识符，则验证数字签字；

如果软件应用程序不包括真实的数字签字，则拒绝软件应用程序访问一个或多个 API；

其中，如果签字标识与签字标识符对应,则验证数字签字的步骤包括：

验证与签字标识符对应的签字标识；

把公用签字密钥存储到移动设备上，该公用签字密钥对应与代码签字授权机构关联的专用签字密钥，根据软件应用程序的杂乱信号，所述代码签字授权机构产生数字签字；

产生软件应用程序的杂乱信号，以获得产生的杂乱信号；

把公用签字密钥应用到数字签字，以获得恢复的杂乱信号；

将产生的杂乱信号与恢复的杂乱信号进行比较。

按照本发明的另一方面，一种控制软件应用程序访问具有签字标识符的应用编程接口 API 的方法，包括：

验证由各个软件应用程序提供的数字签字，以访问 API，其中，签字标识对应各个 API 的签字标识符，且根据对应访问至少一个 API 的签字标识符的签字标识产生用于软件应用程序的数字签字；

如果软件应用程序提供的数字签字是真实的，允许访问至少一个 API；

软件应用程序的数字签字和签字标识由代码签字授权机构产生；

其中，通过将专用签字密钥应用到软件应用程序的杂乱信号产生数字签字；

通过产生软件应用程序的杂乱信号来获得产生的杂乱信号、并将公用签字密钥应用到数字签字中来获得恢复的杂乱信号、验证产生的杂乱信号和恢复的杂乱信号是否相同来验证数字签字。

附图说明

图 1 是根据本发明实施例的代码签字协议图；

图 2 是图 1 的代码签字协议的流程图；

14

图 3 是在移动设备上的代码签字系统方框图；

图 3A 是在一组移动设备上的代码签字系统方框图；

图 4 是图 3 和图 3A 代码签字系统的工作流程图；

图 5 是管理图 3A 的代码签字真实性的流程图；

图 6 是移动通信设备的方框图，其中可实现代码签字系统和方法。

具体实施方式

图 1 是本发明一个实施例的代码签字协议图。应用程序开发商 12 产生软件应用程序 14（应用程序 Y），用于要访问移动设备上一个或多个敏感的 API 的移动设备。软件应用程序 Y14 可以是 Jara 应用程序，它工作于安装在移动设备中的 Java 虚拟机。API 能使软件应用程序 Y 与应用平台界面连接，该应用平台可包括如设备硬件、操作系统、核心软件和数据模块这样的资源。为了调用或与这些设备资源交互作用，软件应用程序 Y 必须访问一个或多个 API，因此 API 可有效地"桥接"软件应用程序和有关的设备资源。在本说明和附着的权利要求中，涉及 API 访问应理解包括以这样方法访问 API，即允许软件应用程序 Y 与一个或多个相应设备资源交互作用，因此，在提供访问任何 API 的同时，允许软件应用程序 Y 与有关的设备资源交互作用，而否定访问 API，则防止软件应用程序与有关资源交互作用。例如，数据库 API 可与设备文件或数据储存系统通信，访问数据库 API 将提供软件应用程序 Y 与文件或数据存储系统之间交互作用。用户界面（UI）API 可与控制器和／或控制软件通信，用于像屏幕、密钥盘、和任何其它向用户提供输出或从用户接收输入的设备部件。在移动设备中，无线电 API 也可作用界面提供给无线通信资源，例如发射机和接收机。同样，加密的 API 可提供与保密模块交互作用，后者在设备上实现保密运算。这些仅仅是可在设备上提供 API 的例子。设备可包括任何这些例子的 API，或不同的 API 代替或附加到上面所述的例子中。

可取的是，任何 API 可分类成由移动设备制造商、或由 API 作者，无线网络工作者，设备拥有或操作者敏感的，或其它实体理解的，后者可由在设备软件应用程序中的病毒或病毒码影响。例如，移动设备制造商可分成对加密程序，无线通信功能或专用的数据模型（如地址薄或日历本）

15

互作用敏感。为防备无授权情况下对这些敏感的 API 访问，要求应用程序
开发商 12 从移动设备制造商获得一个或多个数字签字，或从其它按敏感
分类任何 API 的实体中获得一个或多个数字签字，或从影响到制造商利益
的代码签字授权机构或其它有意保护访问敏感的设备 API 的实体获得数
字签字，并把签字添加到软件应用程序 Y14。

在一个实例中，对每个要访问的敏感的 API 或包括 API 的程序库获
得数字签字。在某些情况下，需要多个签字，这就允许服务提供商，公司
或网络工作者限制某些或全部软件应用程序在特定的一组移动设备上加
载或更新。在这一多签字方案中，所有 API 被限制和锁定，直到对软件应
用程序的"全局"签字得到验证。例如，公司可能希望防止它的顾员在没
有首先获得公司信息技术（IT）或计算机服务部准许的情况下，在它们的
设备上运行任何软件应用程序，于是所有这些公司的移动设备可构成在软
件应用程序能被执行前，至少需要全局签字，即使要访问敏感的 API 和程
序库，根据相应数字签字的验证，作出进一步限制。

二进制可执行的软件应用程序 Y 的表达可与具体的移动设备类型或
移动设备型号无关。软件应用程序 Y14 可以是一次写入任何地方可运行
的二进制格式，与 Java 软件应用程序的情况一样。但是，可能要对每种
移动设备类型或型号有数字签字，或代以对每种移动设备平台或制造商有
数字签字。因此，如果软件应用程序把几种移动设备作为对象的话，软件
应用程序 Y14 可送请几个代码签字授权机构。

软件应用程序 Y14 从应用程序开发商 12 送到代码签字授权机构 16。
在图 1 所示的实施例中，代码签字授权机构 16 检查软件应用程序 Y14，
如在下面更详细描述那样，设想代码签字授权机构 16 也可以或代替考虑
应用软件开发商 12 的身份，以确定是否应对软件应用程序签字。代码签
字授权机构 16 优先地是一个或多个来自移动设备制造商，任何敏感的 API
的作者的代表，或其它具有操作敏感的 API 知识的人（该 API 是软件应
用程序需访问的对象）。

如果代码签字授权机构 16 确定软件应用程序可访问敏感的 API 并因
而要签字，那么对软件应用程序的签字（未画出）由代码签字授权机构
16 产生并附加软件应用程序 Y14。然后，经签字的软件应用程序 Y22，

16

包括软件应用程序 Y14 和数字签字，返回应用程序开发商 12，数字签字
优先地是一标签，它是用只有代码签字授权机构 16 保持的专用签字密钥
18 产生。例如，根据一种签字方案，用 hash 算法（如保密杂乱信号（hash）
算法 SHAI）可产生软件应用程序 14 的杂乱信号（hash），然后与专用的
签字密钥 18 一起用，以建立数字签字。在某些签字方案中，专用签字密
钥用于加密要签字的信息的杂乱信号（hash），例如软件应用程序 Y14，而
在其它方案中，专用密钥可以其它方式用于从要签字的信息或该信息的变
换版本产生签字。

　　然后，把经签字的软件应用程序 Y12 发送给移动设备 28 或由移动设
备 28 在无线网络 24 上下载，但应当理解，本发明的代码签字协议不限于
在无线网上下载的软件应用程序，例如，在另一实施例中，经签字的软件
应用程序 Y22 可通过计算机网络下载到个人计算机，并通过串联连接加
载到移动设备，或可以任何其它形式从应用程序开发商 12 获得并加载到
移动设备上。一旦经签字的软件应用程序 Y22 装到移动设备 28 上，每一
数字签字，优先用公司签字密钥 20，在软件应用程序 Y14 准许访问敏感
的 API 程序库之前，进行验证。虽然经签字的软件应用程序 Y22 装在设
备上，但应理解，即使在设备上可执行的软件应用程序是软件应用程序
Y14。如前面所述，经签字的软件应用程序 Y22 包括软件应用程序 Y14
和一个或多个附加的数字签字（未示出）。当签字被验证时，软件应用程
序 Y14 可在该设备上执行并访问已验证相应签字的任何 API。

　　公用签字密钥20相应于由代码签字授权机构16保持的专用签字密钥
18，并且优先与敏感的 API 一起安装在移动设备上。但是，公用密钥 10
可用设备 28 或可能的个人计算机系统替换从公用密钥库获得（未示出），
并按需要安装在设备 28 上。根据签字方案的一个实施例，移动设备 28 计
算经签字的软件应用程序 Y22 中的软件应用程序 Y14 的杂乱信号（hash），
其中使用与代码签字授权机构 16 相同的散列算法，并用数字签字和公用
签字密钥 20 来恢复由签字授权机构 16 计算的杂乱信号（hash），然后把
本地算得的杂乱信号（hash)结果与从数字签字恢复的杂乱信号（hash)进
行比较，如果杂乱信号（hash)相同，则签字被验证。于是，软件应用程
序 Y14 可能在设备 28 上执行，并访问相应签字已被验证的敏感的 API。

17

如上所述，本发明决不限于这具体说明签字方案的例子，其它签字方案，包括公用密钥签字方案，也可结合这里描述的代码签字方法和系统使用。

图2是参考图1的上述代码签字协议的流程图30。协议从步骤32开始，在步骤34，软件开发商为需要访问敏感的API或阵列敏感的API的程序库（API程序库A）的移动设备写软件应用程序Y。如上所述，移动设备上的一些或全部API可合成敏感性一类，这样，任何软件应用程序对它的访问都需要数字签字验证，例如软件应用程序Y。在步骤36中，应用程序Y由软件开发商优先使用设备模拟器来测试，该模拟器中，数字签字验证功能已不适用。这样，软件开发商可在从代码签字授权机构获得数字签字之前调试软件应用程序Y。一旦软件应用程序Y写好并调试完毕，则可在步骤38传送给代码签字授权机构。

在步骤40和42，代码签字授权机构检查软件应用程序Y，以确定是否应允许访问敏感的API，并作出接受或拒绝该软件应用程序的决定。代码签字授权机构可应用一组准则来确定是否准许软件应用程序访问敏感的API，包括，例如软件应用程序的大小，由API访问的设备资源，软件应用程序的实用性，与其它软件应用程序的相互作用，包含病毒或其它破坏性的代码，和开发商是否有合同义务或与移动设备制造商有其它业务安排。更多管理代码签字授权机构和开发商的细节，参考图5描述如下。

如果代码签字授权机构接受软件应用程序Y，那么在步骤46，数字签字，最好是签字标识，附加到软件应用程序Y中。如上所述，数字签字可用软件应用程序Y的杂乱信号（hash)和专用签字密钥18来产生。签字标识参考图3和4描述如下。一旦数字签字和签字标识加到软件应用程序Y，得到签了字的软件应用程序，则经签字的软件应用程序在步骤48返回软件开发商。然后，软件开发商可申请把签字的软件应用程序Y装到移动设备（步骤50）上的许可证。如果代码签字授权机构拒绝软件应用程序Y，那么把拒绝说明发送给软件开发商（步骤44），软件应用程序Y将不能访问与该签字有关的任何API。

在另一个实施例中，软件开发商可提供软件应用程序Y的杂乱信号（hash)给代码签字授权机构，或以某种简化的格式提供软件应用程序Y。如果软件应用程序是Java应用程序，那么设备有关的二进制*.class文件

18

可用于杂乱信号（hash)工作中，不过，当软件应用程序想要在特别设备或设备类型上工作时，由本申请的代理人所用的设备有关的文件，例如*.coa 可代替用于杂乱信号（hash)或其它数字签字工作中。借助于只提供软件应用程序 Y 的杂乱信号（hash)或简化版本，软件开发商可把没有显示专有代码签字的软件应用程序给代码签字授权机构。软件应用程序 Y 的杂乱信号（hash)与专门的签字密钥 18 一起，可用来由代码签字授权机构产生数字签字。如果其它简化的软件应用程序 Y 的版本发送给代码签字授权机构，那么该简化的版本同样可用来产生数字签字，只要简化的方案或算法，像杂乱信号（hash)算法一样，对不同的输入产生不同的输出。这就保证了每个软件应用程序可有不同的简化版本和因此不同的签字，该签字只能在附加到产生简化版本的具体相应的软件应用程序时才能验证。因为这一实施例不能使代码签字授权机构对病毒或其它破坏性代码来充分评审软件应用程序，因此，也可要求软件开发商和代码签字授权机构之间进行登记处理。例如，代码签字授权机构可预先同意可信任的软件开发商访问一组有限的敏感的 API。

在另一个实施例中，软件应用程序 Y 可提交给多于一个签字机构，每个签字机构可负责对特定敏感的 API 或特定型号的移动设备上的 API 或支持由软件应用程序要求的敏感的 API 的移动设备组的软件应用程序的签字。制造商，移动通信网络操作员，服务商，或公司用户可对使用敏感的 API 有签字权，用于他们特定的移动设备型号，或工作于特定网络上的移动设备，预订一个或多个具体业务，或分配到公司顾员。经签字的软件应用程序可包括软件应用程序和至少一个来自每个签字机构的附加数字签字。尽管这些签字机构在本例中能对同样软件应用程序产生签字，但不同的签字和签字验证方案可与不同的签字机构有关。

图 3 是移动设备 62 上代码签字系统 60 的方框图。该系统 60 包括虚拟机 64，一组软件应用程序 66－70，一组 API 程序库 72－78，和应用平台 80。应用平台 80 最好包括所有移动设备 62 上的资源，它们可由软件应用程序访问。例如，应用平台可包括设备硬件 82，移动设备操作系统 84，或核心软件和数据模型 86。每个 API 程序库 72－78 最好包括一组 API，它与应用平台中的有效资源接口，例如，一个 API 程序库可包括所

19

有与日历程序和日历项数据模型接口的 API。另一个 API 程序库可包括所有与移动设备 62 的传输线路和功能接口的 API。再另一个 API 程序库可包括所有能与移动设备操作系统 84 执行的低级业务接口的 API。此外，一组 API 程序库 72－78 既可包括阵列敏感的 API 74 和 78 的程序库，例如与保密功能的接口，也可包括可被访问而没有阵列敏感的 API 的程序库 72 和 76。同样，一组软件应用程序 66－70 既可包括签字的软件应用程序 66 和 70，它们要求访问一个或多个敏感的 API，也可包括未签字的软件应用程序，如 68。虚拟机 64 优先地是面向运行时环境的目标，如 Sun Micro 系统的 J2ME™（Java2 平台，Micro 出版），它管理移动设备 62 上工作的所有软件应用程序 66－70，并把软件应用程序 66-70 链接到各 API 程序库 72－78。

软件应用程序 Y70 是经签字的软件应用程序的例子，每个经签字的软件应用程序优先包括实际的软件应用程序，如包括能在应用平台 80 上执行的软件代码的软件应用程序 Y，一个或多个签字标识 94 和一个或多个相应的数字签字 96。在签字的软件应用程序 66 或 70 中，每一数字签字 96 和相应的签字标识 94 相应于敏感的 API 程序库 74 或 78，它是软件应用程序 X 或软件应用程序 Y 要求访问的 API。敏感的 API 程序库 74 或 78 可包括一个或多个敏感的 API。在一个替换的例子中，签字的软件应用程序可包括数字签字 96，用于在 API 程序库 74 或 78 中的每个敏感的 API。签字标识 94 可以是唯一的整数，或某些把数字签字 96 与特定 API 程序库 74 或 78、API、应用平台 80 或移动设备 62 的型号相连系的其它装置。

API 程序库 A78 是阵列敏感的 API 的 API 程序库的例子。每个包括敏感的 API 的 API 程序库 74 和 78 应优先包括描述字符串 88，公用签字密钥 20，和签字标识符 92。签字标识符 92 优先相应于签字的软件应用程序 66 或 70 中的签字标识，并能虚拟机让数字签字 96 与 API 程序库 74 或 78 快速匹配。公用密钥 20 相应于由代码签字授权机构保持的专用签字密钥 18，并用于验证数字签字 96 的真实性。描述字符串 88 可以是文本消息，当加载签字的软件应用程序时，它显示在移动设备上，或换句话说，当软件应用程序 X 或 Y 要想访问敏感的 API 时，它显示在移动设备上。

操作上，当签字的软件应用程序 68－70（分别包括要访问敏感的 API

20

程序库 74－78 的软件应用程序 X，Z，或 Y）装到移动设备上时，虚拟机 64 搜索附加的、与 API 程序库 74 或 78 有关的数字签字 96 的符号。优先地，由虚拟机 64 借助于把 API 程序库 74 或 78 中的签字标识符 92 与签字的软件应用程序中的签字标识 94 相匹配而测出合适的数字签字 96。如果签字的软件应用程序包括合适的数字签字 96，那么，虚拟机 64 用公用密钥 20 验证其真实性，然后，一旦合适的数字签字 96 被测出并验证，在执行软件应用程序 X 或 Y 并访问敏感的 API 之前，则描述字符串 88 显示在移动设备上。例如，描述字符串 88 可显示这样的消息"应用程序 Y 要想访问 API 程序库 A"，并借助向移动设备用户提供批准或否定访问敏感的 API 的最后控制。

　　图 3A 是在一组移动设备 62E，62F 和 62G 上的代码签字系统 61 的方框图。系统 61 包括一组移动设备，其中只有三个 62E，62F 和 62G 示于图中。还示出了签字的软件应用程序 70，它包括软件应用程序 Y，两个相应于签字标识 94E 和 94F 的数字签字 96E 和 96F 已加到该软件应用程序上。在作为例子的系统 61 中，由数字签字和标识组成的每对 94E／96E 和 94F／96F，相应于移动设备 62 的型号、API 程序库 78 或有关的平台 80。如果签字标识 94E 和 94F 相应于移动设备 62 的不同型号，那么，当签字的软件应用程序 70，它包括要访问敏感的 API 程序库 78 的、经签字的软件应用程序 Y 装到移动设备 62E 上时，虚拟机 64 借助于把标识 94E 与签字标识符 92 相匹配来为与 API 移动库 78 有关的数字签字 96E 搜索签字的软件应用程序 70。同样，当签字的软件应用程序 70，它包括要访问敏感的 API 程序库 78 的软件应用程序 Y，装到移动设备 62 上时，在设备 62F 中的虚拟机 64 为与 API 程序库 78 有关的数字签字 96F 搜索软件应用程序 70。但是，在要访问敏感的 API 程序库 78 的、经签字的软件应用程序 70 中的软件应用程序 Y 装到应用程序开发商未获得数字签字的移动设备的型号上时，图 3 中的设备 62G，设备 64G 中的虚拟机 64 找不到附加于软件应用程序 Y 的数字签字，因此否定在设备 62G 上访问 API 程序库 78。从前面描述应可以理解，像软件应用程序 Y 那样的软件应用程序可以有多个规定的设备，规定的程序库，或规定的 API 签字或加于其上的这些签字的组合。同样，对不同的设备构成不同的签字验证要求，例如，

21

设备 62E 可要求既有全局签字，又有对任何敏感的 API 的附加签字，为了使该软件应用程序得以执行，软件应用程序需访问 API。而设备 62F 可要求只有全局签字的验证，设备 62G 可要求只对其敏感的 API 签字的验证。很明显，通信系统可包括装置（未示出），在该装置上，接收的作为如 70 的签字的部分软件程序的软件应用程序 Y 可以执行而没有任何签字验证。虽然签字的软件应用程序有一个或多个附加的签字，但软件应用程序 Y 可能在某些设备上执行而没有首要的任何签字验证。对软件应用程序的签字最好不与它在没有实现签字验证的设备上的执行相干涉。

图 4 是流程图 100，表示图 3 和图 4 的代码签字系统的工作。在步骤 102，软件应用程序装到移动设备上，一旦软件应用程序安装完毕，该设备最好用虚拟机来确定该软件应用程序是否要访问任何阵列敏感的 API 的 API 程序库（步骤 104）。如果否，那么软件应用程序与所有它所要求的 API 程序库连接并执行（步骤 118），如果软件应用程序要访问敏感的 API，那么在步骤 106－116 中，虚拟机验证该软件应用程序包括与任何要访问的敏感的 API 有关的有效数字签字。

在步骤 106，虚拟机从敏感的 API 程序库查找公用签字密钥 20 和签字标识符 92，签字标识符 92 被虚拟机在步骤 108 中用来确定软件应用程序是否有附加的数字签字与相应的签字标识 94 相应。如果没有，则软件应用程序没有被代码签字授权机构批准访问敏感的 API，并最好防止软件应用程序在步骤 116 中执行。在另一个实例中，没有合适数字签字 96 的软件应用程序可以移动设备上消除，或可以否定它访问阵列敏感的 API 的 API 程序库，但可在没有访问 API 程序库的可能范围内执行。也可想到，当签字验证失效时，用户可以有输入提醒，供用户控制后续操作从设备中消除该软件应用程序。

如果相应于敏感的 API 程序库的数字签字 96 加到软件应用程序并由虚拟机测出，那么，虚拟机用公用密钥 20 来验证该数字签字 96 的真实性（步骤 110）。这一步可用上面描述的签字验证方案或其它替换的签字方案来执行。如果数字签字 96 不真实，则软件应用程序最好不被执行、消除或如上所述限制访问敏感的 API（参考步骤 116）。如果数字签字是真实的，则描述字符串 88 最好在步骤 112 中显示，警告移动设备用户，该软

22

件应用程序要访问敏感的 API，并提示用户授权执行或安装该软件应用程序（步骤 114）。当软件应用程序有多于一个签字要验证时，在 112 步提示用户之前，最好对每一签字重复步骤 104－110。如果步骤 114 中的移动设备用户认可该软件应用程序，则它可被执行并连到敏感的 API 程序库（步骤 118）。

图 5 是流程图，表示图 3A 的代码签字授权机构的管理 200。在步骤 210，应用程序开发商已开发了新的软件应用程序，它要在一个或多个目标设备型号或类型上执行。目标设备可包括来自不同制造商的一组设备，来自同一制造商的一组设备模型或类型，或一般具有特别签字和验证要求的任一组设备。"目标设备"一词涉及有共同签字要求的设备。例如，对执行所有软件应用程序要求全局签字的一组设备可包括目标设备。既要求全局签字又要求对敏感的 API 的进一步签字的设备可以是多于一个目标设备组的部分。软件应用程序可用至少一个已知的 API 以与设备无关的状态写成，可在至少一个有 API 程序库的目标设备上获得支持。最好是，被开发的软件应用程序要在几个目标设备上执行，其中每个至少有它自己的一个 API 程序库。

在步骤 220，对一个目标设备的代码签字授权机构从开发商接收目标签字请求，目标签字请求包括软件应用程序或软件应用程序的杂乱信号（hash），开发商标识符，以及至少一个目标设备标识符，它识别请求签字的目标设备。在步骤 230，签字机构查阅开发商数据库 235 或其它记录，以确定是否信任开发商 220。这一确定可根据前面讨论的几个准则来做，例如开发商是否有合同义务或已进入设备制造商，网络工作者，服务供应商安排的某些其它类型的业务。如果开发商是可信的，则该方法在步骤 240 开始。但是，如果开发商不可信，则该软件应用程序被拒绝（250），并不被签字机构签字。假定开发商是可信任的，则在步骤 240，签字机构借助于查询专用密钥存储器，如目标专用密钥数据库来确定它是否有相应于提交的目标标识符的目标专用密钥 245，如果找到目标专用密钥，则在步骤 260 产生对该软件应用程序的数字签字，并且该数字签字或经签字的软件应用程序（包括附加到该软件应用程序的数字签字）返回开发商（步骤 280）。但是，如果目标专用密钥在步骤 240 没有找到，则该软件应用

23

程序在步骤 270 被拒绝，并不对该软件应用程序产生数字签字。

方便的是，如果目标签字机构接受图 5 方法得可兼容的实例，则为了方便管理代码签字授权机构和开发商共同体代码签字过程，可建立目标签字机构的网络，以便对多个具有毁坏码的低似然性的目标提供经签字的软件应用程序。

当软件应用程序在设备上执行时，一经发现或根据其表现怀疑软件应用程序中有任何破坏性或其它有问题的码，那么，相应的应用程序开发商与任何或全部签字机构的登记或特权可被怀疑或取消，因为数字签字提供了检查跟踪，通过它可识别有问题的软件应用程序的开发商。在这种事件中，设备者借助于配置周期性下载签字取消表通知取消。如果相应的数字签字已被取消的软件应用程序在设备上运行，那么该设备可停止任何这种软件应用程序的执行，并合理地从其本地存储器中消除。如果愿意，设备还可配置重新执行签字验证，例如周期性地或当新的取消表被下载时。

虽然由签字机构产生的数字签字与应用程序开发商的身份验证和确认该应用程序开发商已确实注册，那么数字签字优先从软件应用程序的杂乱信号（hash)或其它变换的版本产生，并成为专门的应用，这与已知的代码签字方案不同，其中允许任何来自可信的应用程序开发商或作者的软件应用程序访问 API。在这里描述的代码签字系统和方法中，API 的访问是逐个应用的基础上准许的，因而能比较严格地控制或限制。

图 6 是移动通信设备的方框图，其中可实现代码签字系统和方法。移动通信设备 610 最好是双程通信设备，它至少具有声音和数据通信能力。该设备优先具有与互联网上的其它计算机系统通信的能力。根据由设备提供的功能，设备可称为数据收发设备，双程寻呼机，有数据收发功能的蜂窝电话，无线互联网设备或数据通信设备（带或不带电话功能）。

在设备能用于双程通信的地方，设备将采用通信分系统 611，它包括接收机 612，发射机 614，和有关的一个或多个嵌入的或内部的部件，天线单元 616 和 618，木地振荡器（LO）613，和处理模块，例如数字信号处理器（DSP）620。通信领域内的业务人士知道，通信系统 611 的具体设计与设备要在其中工作的通信网络有关。例如，北美市场用的设备 610 可包括通信分系统 611，它设计成在 Mobitex™ 移动通信系统或 DataTAC™

24

移动通信系统内工作，而用于欧洲的设备 610 可采用通信分组无线业务
（GPRS）通信分系统 611。

网络访问要求也随网络 919 的类型而变化，例如，Mobitex 和 DataTAC
网络中，移动设备 610 用与每个设备有关的唯一识别数字在网上注册，但
在 GPRS 网络中，网络访问与设备 610 的用户有关。因此，GPRS 设备为
在 GPRS 网上工作要求用户识别模块（未示出）。通常称为 SIM 卡。没有
SIM 卡，GPRS 设备将不能起充分的作用。本地或无网络通信功能（如果
有）可以运作，但设备 610 不能在网络 619 上实行任何功能，包括通信，
除了像"911"紧急呼叫那样合法地所要求的工作。

当要求的网络注册或激励过程已完成时，设备 610 可在网络 619 上发
送和接收通信信号。由天线 616 通过通信网络 619 收到的信号输入接收机
612，它可实行普通接收机的功能，例如信号放大，下变频，滤波，通道
选择等等，以及在图 6 系统所示的例中的模—数变换。接收信号的模数变
换允许比较复杂的通信功能，例如解调和解码可在 DSP620 中执行。以同
样的状态处理发射信号，包括用 DSP620 调制和编码，并输入发射机 614
作数—模变换，上变频，滤波，放大和通过天线 618 在通信网络 619 上传
输。

DSP620 不仅处理通信信号，也为接收机和发射机提供控制，例如，
作用于接收机和发射机中的通信信号的增益可通过在 DSP620 中实现的自
动增益控制算法进行自适应控制。

设备 610 优先包括微处理机 638，它控制整个设备的工作。通信功能，
至少包括数据和声音通信，通过通信分系统 611 实行。微处理器 638 也与
另外的分系统或资源，如显示器 622，闪存 624，随机访问存储器（RAM）
626，辅助输入／输出（I／O）分系统 628，串口 630，密钥盘 632，扬声
器 634，麦克风 636，短距通信分系统 640 和任何其它的设备分系统（统
称 642）互作用。API，包括敏感的 API，它要求在准许访问前验证一个
或多个数字签字，可安装在设备 610 上，提供软件应用程序上图 6 中的任
何资源的接口。

图 6 中所示的某些分系统执行与通信有关的功能，而其它分系统可提
供"常驻的"或在设备上的功能。要说明的是，某些分系统，例如密钥盘

25

632 和显示器 622，既可用于与通信有关的功能，如输入文本消息用于在通信网络上传输，也可用于常驻设备的功能，如计算器或任务表。

微处理器 638 所用的操作系统软件和由软件应用程序访问的合理的API，优先存入永久性存储器，如闪存 624，它可替代只读存储器（ROM）或类似的存储单元（未示出）。业内人士理解，操作系统，专门的设备软件应用程序，或其中的部分，可临时装到易失性存储器（如 RAM626）中。接收和发射的通信信号也可存入 RAM620。

微处理器 638，除了它的操作系统功能，能优先执行在设备上的软件应用程序。预定的一组应用程序控制基本的设备操作，包括至少数据和声音的通信应用程序，通常在制造期间就装在设备 610 上。可装在设备上的优先应用程序可以是个人信息管理（PIM）应用程序，它具有组织和管理涉及设备用户的数据项目的能力，例如，但不限于电子邮件，日历事件，语音邮件，约定和任务项。自然，在设备上一个或多个存储器是有用的，以适合 PIM 数据项目在设备上储存。这种 PIM 的应用优先具有通过无线网发送和接收数据项的能力。在一个优选实施例中，PIM 数据项通过无线网络无缝连接地集成、合成和更新，以存储的或与主计算机系统有关的设备用户相应的数据项在移动设备上建立关于数据项的镜像主计算机。这对主计算机系统是移动设备用户的办公室计算机系统的情况特别有利。另外的应用软件，包括上述签字的软件应用程序，也可通过网络 619，辅助 I／O 分系统 628，串口 630，短距离通信分系统 640 或任何其它合适的分系统 642 装到设备 610 上。设备的微处理器 638 可验证任何数字签字，包括"全局"设备签字和规定的 API 签字，这些签字在软件应用程序由微处理器 638 执行和／或访问任何有关的敏感的 API 前加到软件应用程序。安装应用程序的这种可塑性增加了设备的功能，并提供增强的在设备功能、有关通信功能或两者。例如，保密通信应用程序可使要用设备 610 通过保密 API 和保密模块（其中实现设备上的保密运算）（未示出）执行的电子商务功能和其它会计事务成为可能。

在数据通信模型中，收到的信号，如下载的文本消息或万维网页，由通信分系统处理并输入微处理器 638，它进一步处理收到的信号，输出到显示器 622，或输出到辅助的 I／O 设备 628。设备 610 的用户也可用密钥

26

盘 632 构成数据项，如电子邮件短文密钥盘 632 是完全的字母数字密钥或电话型的辅助密钥盘，与显示器 622 和合理的 I／O 设备 628 相结合。这样构成的数据项可通过通信分系统 611 在通信网络上传输。

对于声音通信，设备 610 的整体工作基本上是相同，除了收到的信号优先输出给扬声器，发射的信号由麦克风 636 产生之外。可替代的声音或音频 I／O 分系统，例如声音消息记录分系统，也可在设备 610 上实现。虽然声音或音频信号输出主要是通过扬声器 634 完成的，但显示器 622 也可用来提供呼叫方身份，呼叫持续时间，或其它有关信息的语音呼叫。

图 6 中的串口 630 通常是在个人数字助理（PDA）型通信设备中实现的，它可能要与用户桌面计算机（未画）同步，但是一种可选的部件。这种端口 630 使用户能通过外部设备或软件应用程序设置预定选项，并借助于不通过无线通信网络而提供信息或软件下载到设备 610 来扩展设备的能力。这种下载路径可用于把保密密钥直接加载到设备上，这种可靠和可信的连接使保密设备通信成为可能。

短距通信分系统 640 是另一可选的部件，它可提供设备 624 和不同的系统或设备间的通信，合并不需要是同类设备。例如，分系统 640 可包括红外设备和有关的电路及元件，或 Bluetoorh™（蓝牙）通信模式，以提供与有相同能力的系统和设备通信。

这里描述的实施例是相应于权利要求中各部件的结构、系统和方法。本说明可使业内人士能制造和使用相应于权利要求中的可替代的部件。本发明预定的范围包括其它结构、系统或方法，它们与权利要求书的文字语言没有不同，并进一步包括与权利要求书中的文字语言有非实质性判别的结构、系统和方法。

例如，当在图 5 方法中，在步骤 250 拒绝软件应用程序时，签字机构可要求开发商签一合同或与设备制造商或签字机构影响其利益的其它实体建立业务关系。同样，如果在步骤 270 拒绝软件应用程序，对该软件应用程序签字的签字机构可授权给不同的签字机构，这种授权签字基本上可如图 5 所示进行，其中从信任的开发商那里收到最初请求的目标签字机构（步骤 220），根据信任的开发商来自目标签字机构的利益，要求不同的签字机构对该软件应用程序签字。一旦代码签字授权机构间建立起信任关

系，目标专用代码签字密钥可在代码签字授权机构间共享，以改善步骤240方法的性能，或设备可配置成从任何一个信任的签字机构签字。

此外，虽然描述了软件应用程序的上下文，但本发明的代码签字系统和方法也可用于其它设备有关的部件，包括，但不限于，指令和有关的指令变元系统，和构成与设备资源接口的程序库。这种指令和程序库可由设备制造商，设备拥有者，网络工作者，服务提供商，软件应用程序开发商等发送给移动设备。希望根据本权利要求书中描述的代码签字系统和方法，借助于在指令能在设备上执行之前，要求验证一个或多个数字签字，来控制可能影响设备工作的任何指令的执行，例如改变设备标识码或无线通信网络地址的指令。
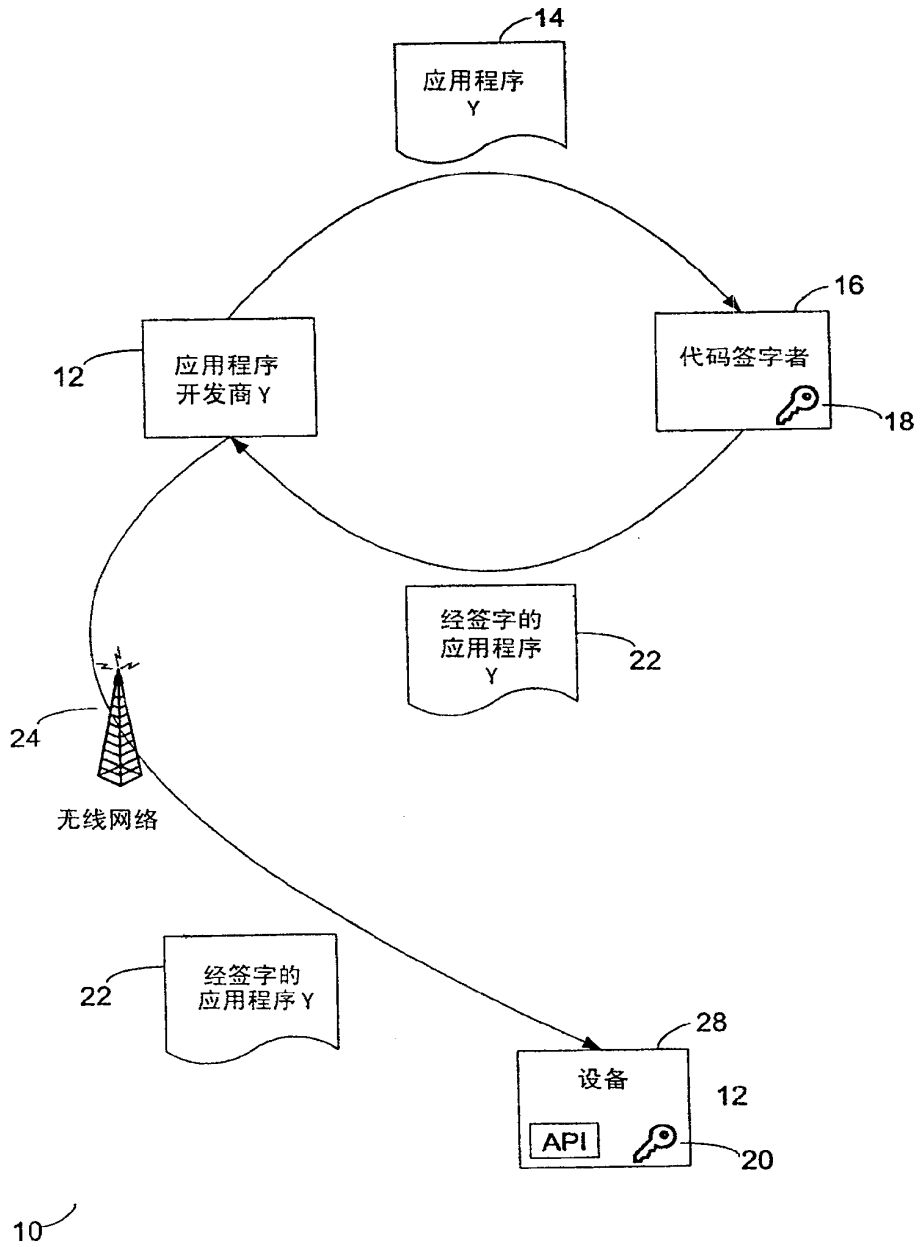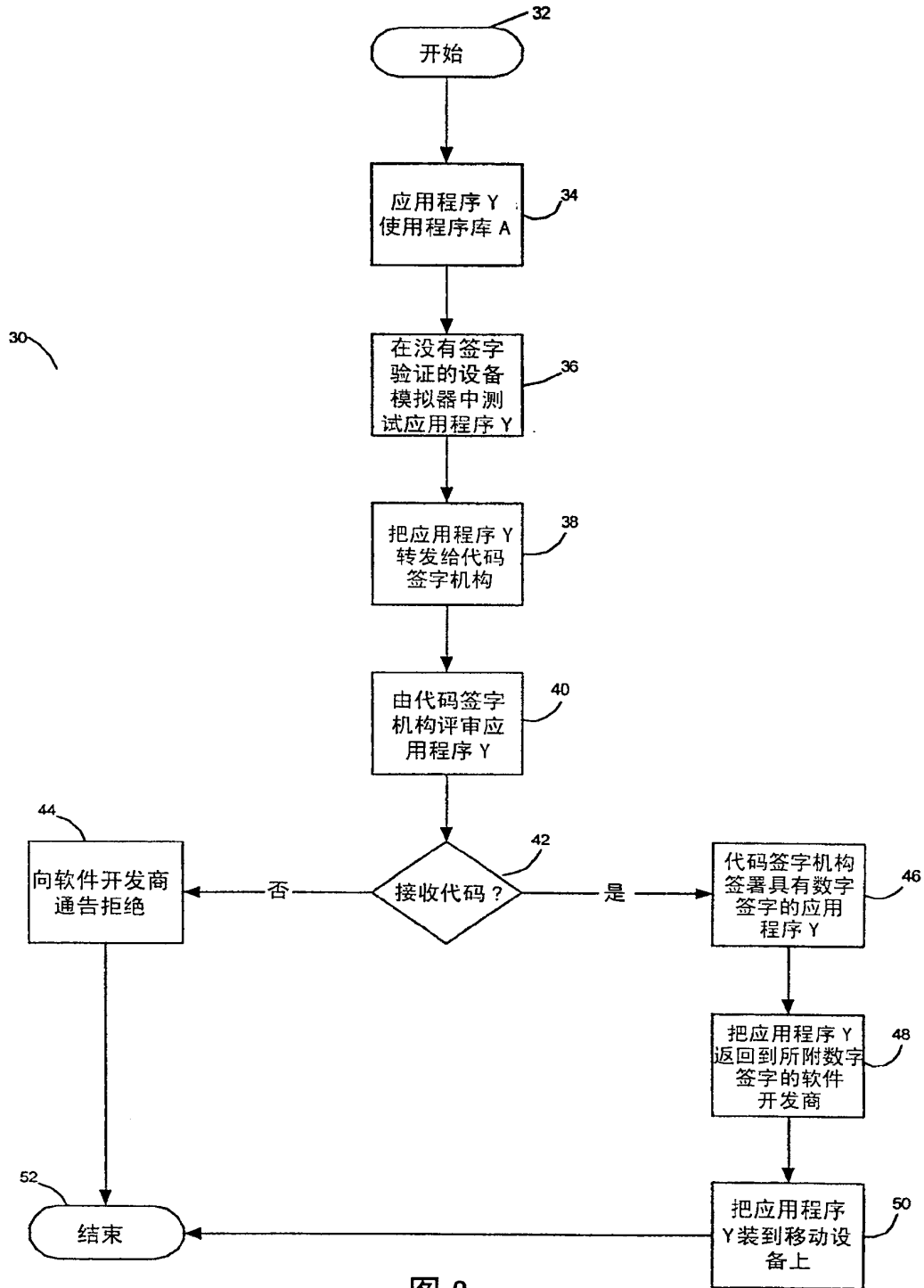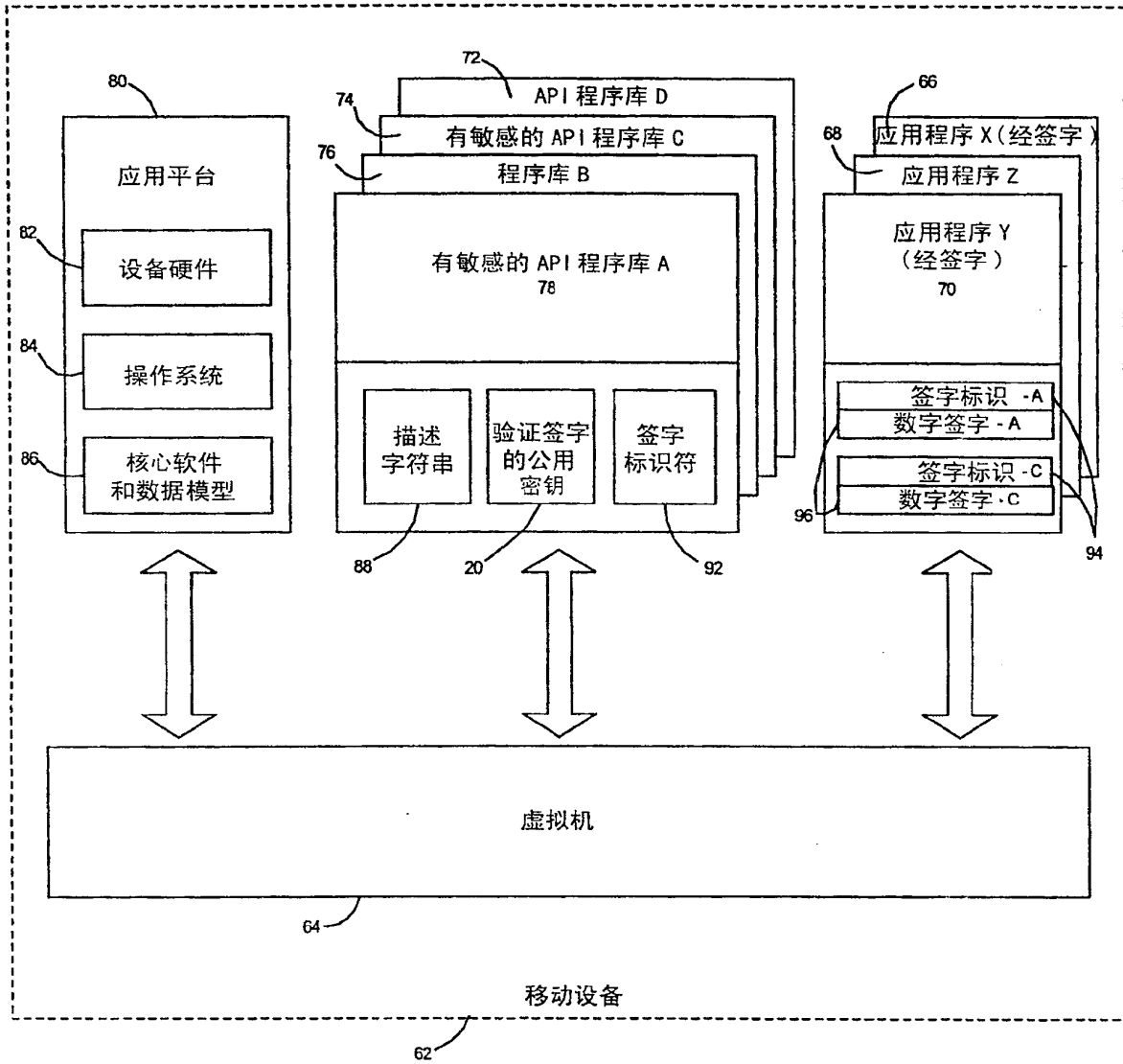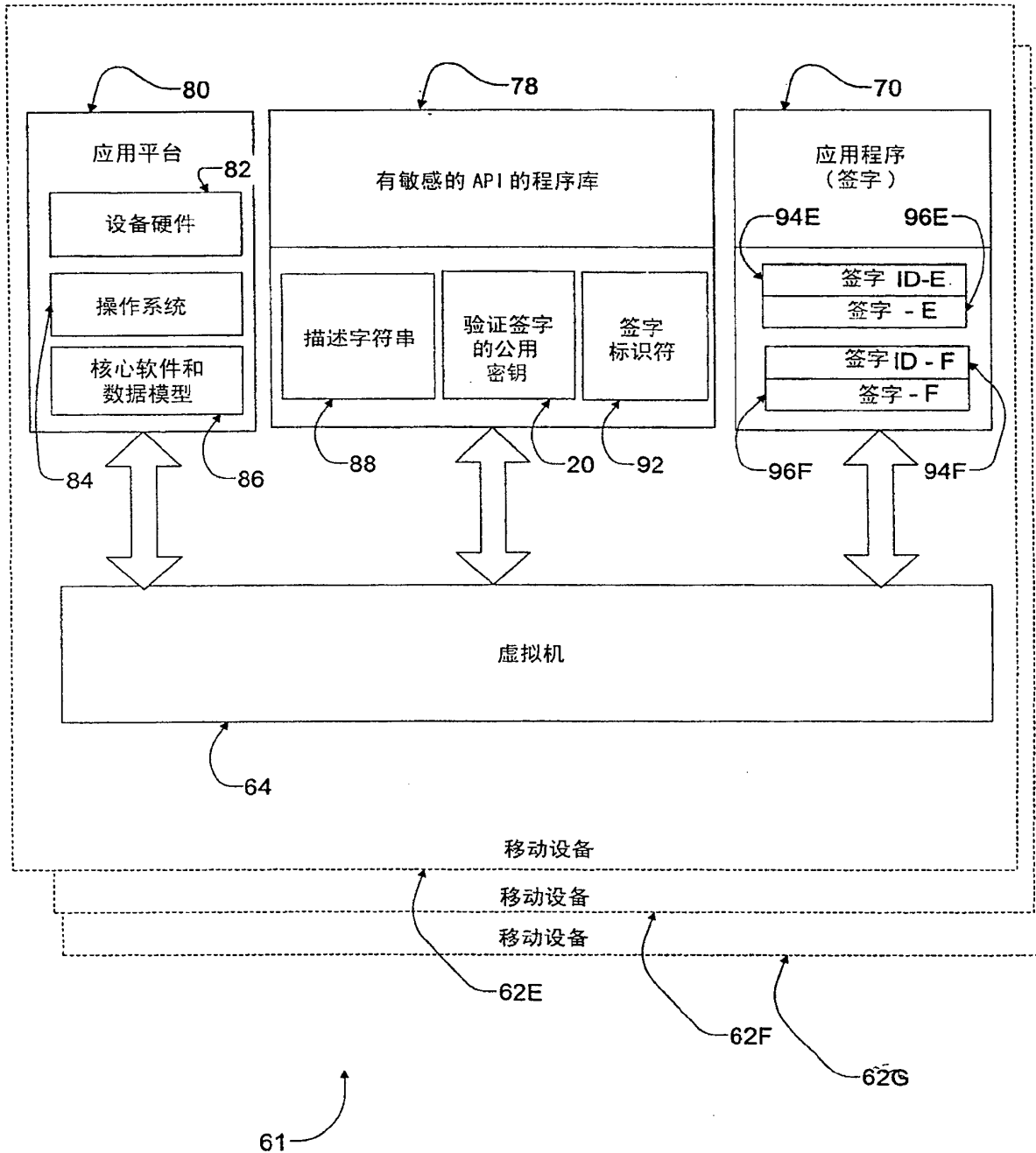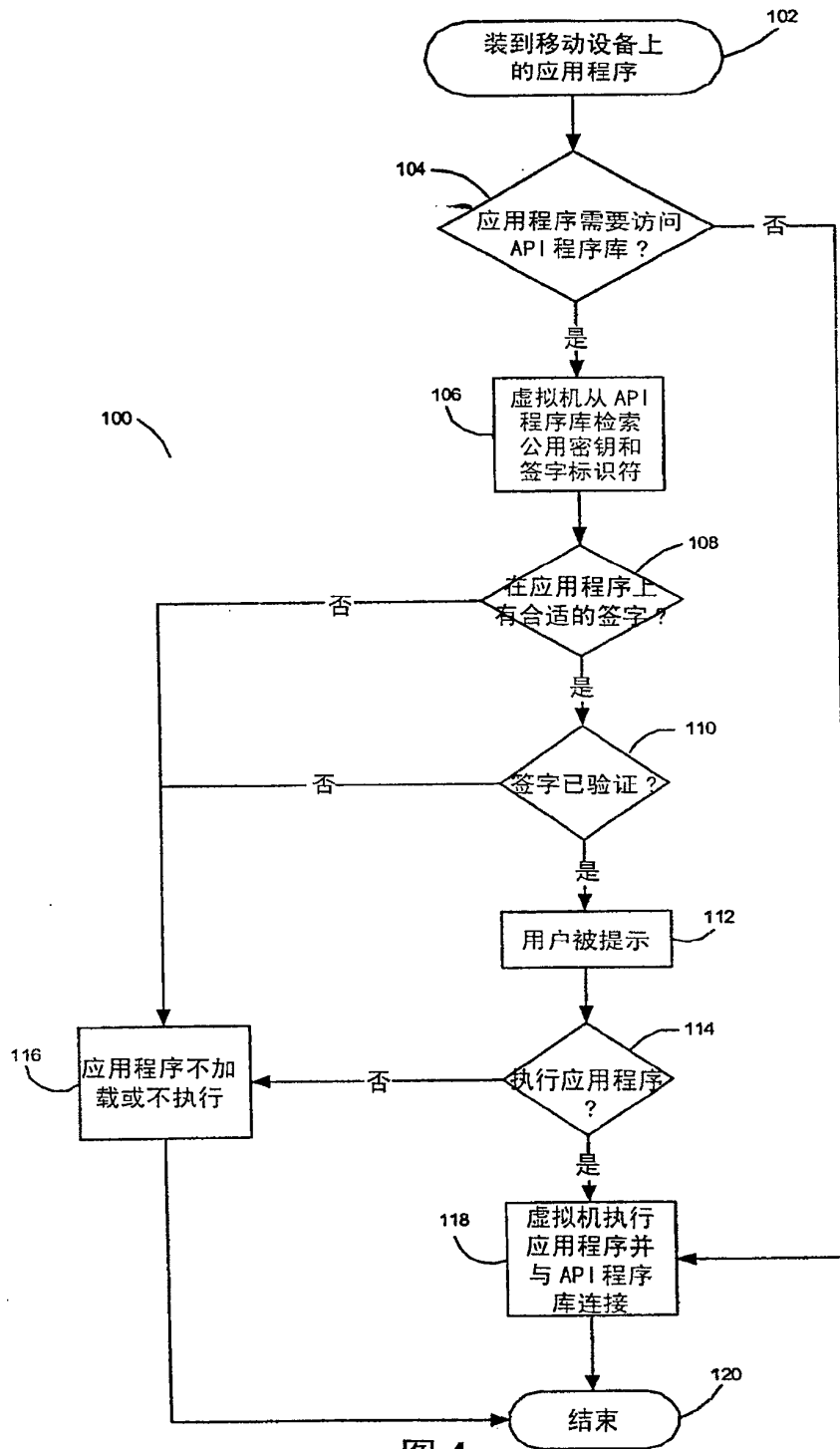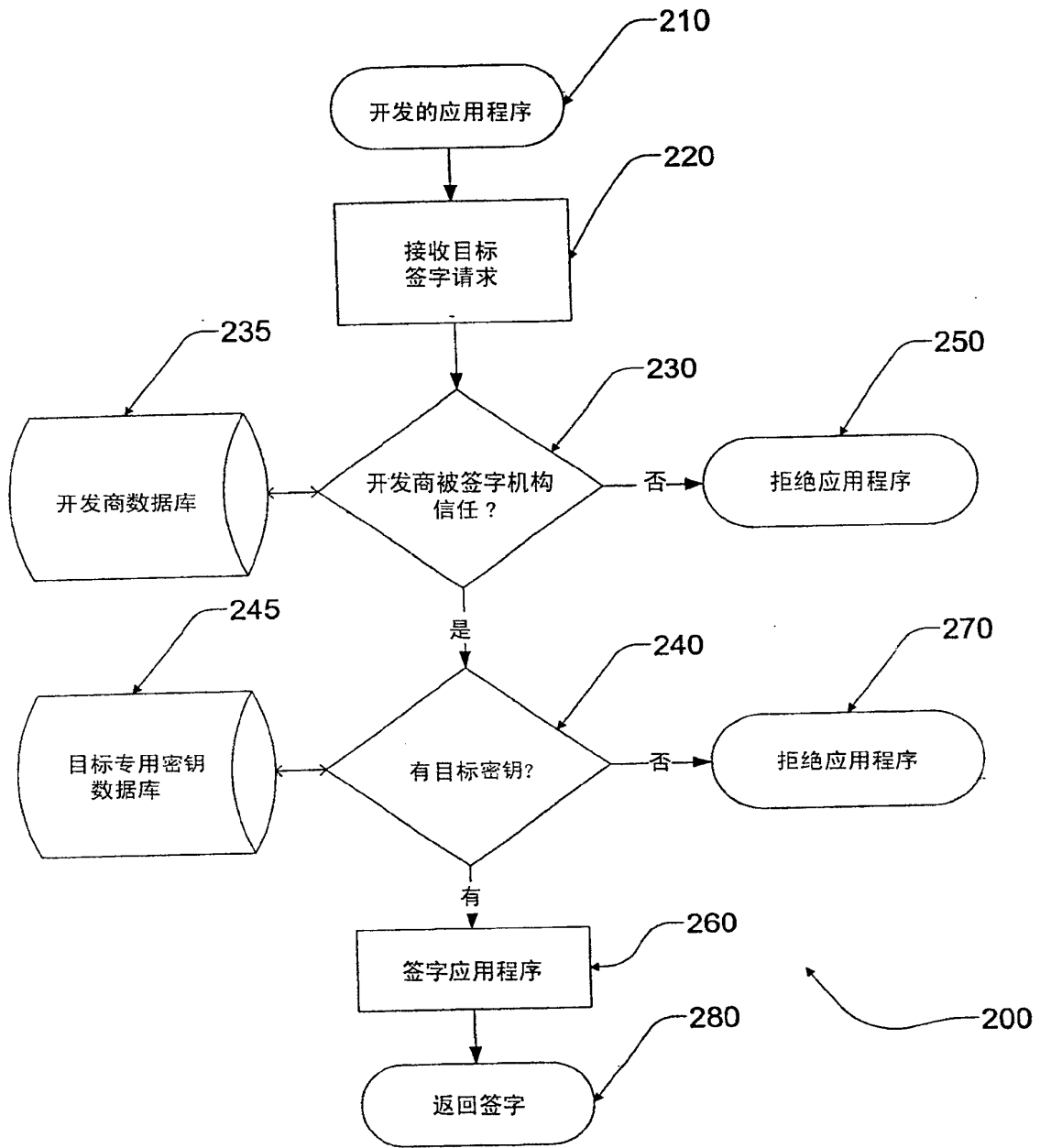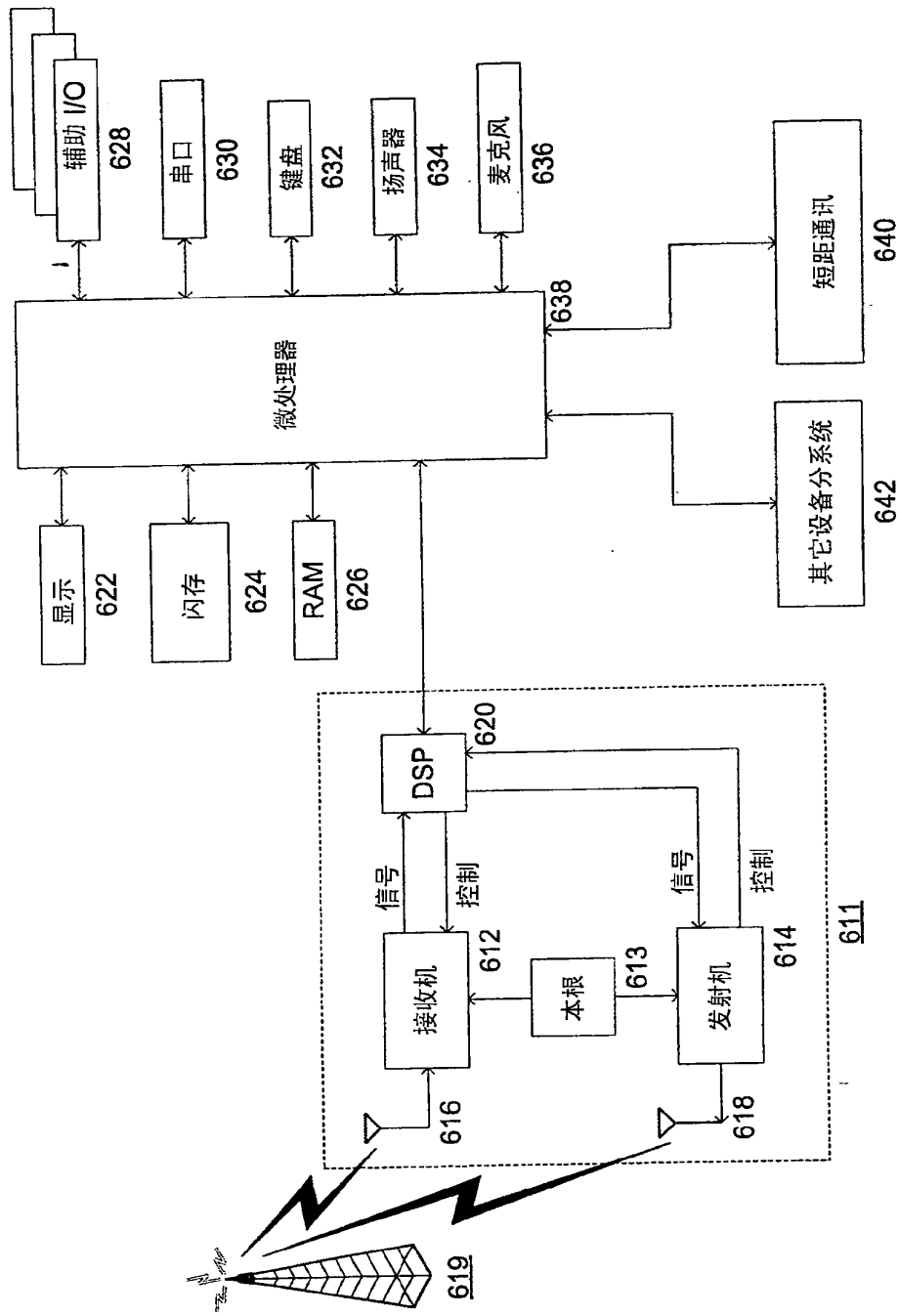
28

图 1

29

```
                              ┌────────┐
                              │  开始  │── 32
                              └────────┘
                                   │
                                   ▼
                          ┌──────────────┐
                          │  应用程序 Y   │── 34
                          │  使用程序库 A  │
                          └──────────────┘
                                   │
                                   ▼
                          ┌──────────────┐
                          │  在没有签字    │── 36
                          │  验证的设备    │
                          │  模拟器中测    │
                          │  试应用程序 Y  │
                          └──────────────┘
                                   │
                                   ▼
   30                     ┌──────────────┐
                          │  把应用程序 Y  │── 38
                          │  转发给代码    │
                          │  签字机构      │
                          └──────────────┘
                                   │
                                   ▼
                          ┌──────────────┐
                          │  由代码签字    │── 40
                          │  机构评审应    │
                          │  用程序 Y      │
                          └──────────────┘
                                   │
                                   ▼
   44                          ◇ 42                         46
 ┌──────────┐              接收代码？         ┌──────────────┐
 │ 向软件开发商 │◄── 否 ──◇        ◇── 是 ──►│ 代码签字机构  │
 │ 通告拒绝    │                              │ 签署具有数字  │
 └──────────┘                                │ 签字的应用    │
      │                                      │ 程序 Y        │
      │                                      └──────────────┘
      │                                             │
      │                                             ▼
      │                                      ┌──────────────┐
      │                                      │ 把应用程序 Y  │── 48
      │                                      │ 返回到所附数字 │
      │                                      │ 签字的软件    │
      │                                      │ 开发商        │
      │                                      └──────────────┘
      │                                             │
   52 │                                             ▼
 ┌────────┐                              ┌──────────────┐
 │  结束  │◄─────────────────────────── │ 把应用程序    │── 50
 └────────┘                              │ Y装到移动设   │
                                         │ 备上          │
                                         └──────────────┘
```

图 2

图 3

31

图 3A

32

图 4

图 5

图 6

[12]

STANDARD PATENT SPECIFICATION
標準專利說明書

[21] Application No. 申請編號

03106586.4

[22] Date of filing 提交日期

12.09.2003

[51] Int.Cl.^{7} G06F

---

[54] SOFTWARE CODE SIGNING SYSTEM AND METHOD 代碼簽名系統和方法

[30] Priority 優先權	[73] Proprietor 專利所有人
21.09.2000 US 234152 P	RESEARCH IN MOTION LIMITED
26.09.2000 US 235354 P	295 PHILLIP STREET
20.02.2001 US 270663 P	WATERLOO
[43] Date of publication of application 申請發表日期	ONTARIO N2L 3W8
16.01.2004	Canada
[45] Publication of the grant of the patent 批予專利的發表日期	[72] Inventor 發明人
04.05.2006	YACH, DAVID, P.
EP Application No. & Date 歐洲專利申請編號及日期	BROWN, MICHAEL, S.
EP 01973901.0 20.09.2001	LITTLE, HERBERT, A.
EP Publication No. & Date 歐洲專利申請發表編號及日期	[74] Agent and / or address for service 代理人及/或送達地址
EP 1320795 25.06.2003	Williams, Davis, Hill & Co
Date of Grant in Designated Patent Office 指定專利當局批予專利	Suite 701, 7th Floor
日期 16.11.2005	6-8 Pottinger Street, Central
	Hong Kong

Electronic Patent Application Fee Transmittal

Application Number:	10381219
Filing Date:	20-Mar-2003
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Filer:	Kendrick Lo./Tobiah Caron
Attorney Docket Number:	555255012423
Filed as Large Entity	

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1615	12	60	720
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension - 3 months with $0 paid	1253	1	1270	1270
Miscellaneous:				
Request for continued examination	1801	1	930	930
Total in USD ($)				**2920**

Electronic Acknowledgement Receipt

EFS ID:	11392216
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Kendrick Lo./Tobiah Caron
Filer Authorized By:	Kendrick Lo.
Attorney Docket Number:	555255012423
Receipt Date:	11-NOV-2011
Filing Date:	20-MAR-2003
Time Stamp:	18:24:31
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$2920
RAM confirmation Number	13206
Deposit Account	022095
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	
Charge any Additional Fees required under 37 C.F.R. 1.492 (National application filing, search, and examination fees)	
Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	10289-US-PCTtransmittal.pdf	59609 63a76fefe276508eefaa810e657d589fae7eede3	no	1
Warnings:					
Information:					
2	Miscellaneous Incoming Letter	10289-US-PCTstatement.pdf	63698 4b4e2cbbbfacb453232cbb32c939fc150d8d88af	no	1
Warnings:					
Information:					
3	Power of Attorney	10289-US-PCTPOA.pdf	331298 4c618c1e8968663d4632db46d29c5690aba40d41	no	2
Warnings:					
Information:					
4	Extension of Time	10289-US-PCTexttime.pdf	57883 3368e34714e797bce8b60e66b13b76ed4b0f3d79	no	1
Warnings:					
Information:					
5	Request for Continued Examination (RCE)	10289-US-PCTRCE.pdf	697268 58f2f5ab1f49f67855507fe7bc0996d9cb3e00ea	no	3
Warnings:					
Information:					
6	Amendment Submitted/Entered with Filing of CPA/RCE	10289-US-PCTOAResponse.pdf	1339618 6136814bacdfb8b5b89ff88e301013d0108fd074	no	28
Warnings:					
Information:					
7	Miscellaneous Incoming Letter	10289-US-PCTIDSltr.pdf	86692 234c4613a4fb1f8c4991a5bb5553afa2458d055d	no	1
Warnings:					
Information:					
8	Information Disclosure Statement (IDS) Form (SB08)	10289-US-PCTIDS.pdf	613217 5fa93875458df9340ca9ec4370a3f2038dc66efc	no	6

Warnings:					
Information:					
A U.S. Patent Number Citation or a U.S. Publication Number Citation is required in the Information Disclosure Statement (IDS) form for autoloading of data into USPTO systems. You may remove the form to add the required data in order to correct the Informational Message if you are citing U.S. References. If you chose not to include U.S. References, the image of the form will be processed and be made available within the Image File Wrapper (IFW) system. However, no data will be extracted from this form. Any additional data such as Foreign Patent Documents or Non Patent Literature will be manually reviewed and keyed into USPTO systems.					
9	Foreign Reference	F1_CN1541350A.pdf	1919961 4ff2dc474301aa85d5b01f20fa717bf99c3ffe58	no	31
Warnings:					
Information:					
10	Foreign Reference	F2_CN101714201A.pdf	1814942 a73591ef5e8a884394884d871fe7528f78afe8f5	no	29
Warnings:					
Information:					
11	Foreign Reference	F3_CN101694688A.pdf	1549685 0658923d7d866611a35856bcba1f5f5038f49a35	no	25
Warnings:					
Information:					
12	Foreign Reference	F4_EP1320795B2.pdf	287750 dcbba963d6f92fab13c37b14f02e8b26a2457fa7	no	23
Warnings:					
Information:					
13	Foreign Reference	F5_EP2306259A2.pdf	401014 c35839d94b09c512c7c90a9c7b2ec22068b79046	no	24
Warnings:					
Information:					
14	Foreign Reference	F6_EP1626324A2.pdf	394488 b334b2904c12f9f88364a0622c0dbe1ce8dbc08c	no	22
Warnings:					
Information:					
15	Foreign Reference	F7_EP2284644A1.pdf	468893 4decc3b8daaa16636abe8c259cb38bfca98b133a	no	24
Warnings:					
Information:					
16	Foreign Reference	F8_EP2278429A1.pdf	455878 5d0ddc4da0a64ad07ad1b29ee741828a4dfed669	no	24
Warnings:					

Information:					
17	Foreign Reference	F9_EP2306260A2.pdf	456662 af3c6d8cc8562194aed22e066499cbc05881 5e0c	no	23
Warnings:					
Information:					
18	Foreign Reference	F10_EP1626325B1.pdf	295884 22bfc10ca4e7cc9962629541264a62f21ad4 a8f4	no	23
Warnings:					
Information:					
19	Foreign Reference	F11_EP1626326B1.pdf	301428 a1454e7e99c114fdf3c55b90b53fb7b37bed 8cd6	no	24
Warnings:					
Information:					
20	Foreign Reference	F12.pdf	237183 4f704fd207291c4a7fdbf6445b1a5a94c5ef1 603	no	3
Warnings:					
Information:					
21	Foreign Reference	F14_HK1091665.pdf	50548 bd4838b5c916dc5d16c34286584cc5d067c 4b987	no	1
Warnings:					
Information:					
22	Foreign Reference	F15_HK1091667.pdf	50496 5c5bff8929b8fe856595a6bfb34a7c9cac061 439	no	1
Warnings:					
Information:					
23	Foreign Reference	F16_100573402C.pdf	1151300 f39895e7c6685caa433632651822bcf3b7c4 2b39	no	35
Warnings:					
Information:					
24	Non Patent Literature	N1_NoticeOfAb.pdf	97611 42865ea8087ae6365fc33cf4cf71a4f8a09ba 60e	no	1
Warnings:					
Information:					
25	Non Patent Literature	N2_CN_OA10Aug11.pdf	680844 79068364d28065cbc42026b4909ab1f9b13 86de2	no	6
Warnings:					

Information:					
26	Non Patent Literature	N3_EESR22June11.pdf	413146 / 19382353ea766a70c1624f87b5ba3322848 3b7e5	no	6
Warnings:					
Information:					
27	Non Patent Literature	N4.pdf	147711 / 0bd086c437f9872f1db9af5fec887ad6de6a aca9	no	2
Warnings:					
Information:					
28	Non Patent Literature	N5_EESR_23Feb11.PDF	128620 / 7eafa319f4638e91b677f50ec90e163d51d4 8fec	no	4
Warnings:					
Information:					
29	Non Patent Literature	N6_EP_OA13Jul11.pdf	206176 / b0e8bdf9423d0cf3a8dd13efc3cce55caa0c 5f1e	no	4
Warnings:					
Information:					
30	Non Patent Literature	N7_EESR_21Dec10.PDF	182893 / 18b7d640b80f2871db43bbccb305ae8519e b22c2	no	5
Warnings:					
Information:					
31	Non Patent Literature	N8_OA_23Feb11.PDF	203426 / 14280269a937e9510109ffd21a4cfb1a8e85 f48f	no	4
Warnings:					
Information:					
32	Non Patent Literature	N9_OA_14Jul11.PDF	196510 / e816a911018612a48dbf153310f078946e9 6590f	no	4
Warnings:					
Information:					
33	Non Patent Literature	N10_EESR22Jul11.pdf	414384 / e346338931a8e4cbda46d2869685f4bf8e5 0f916	no	5
Warnings:					
Information:					
34	Non Patent Literature	N11.pdf	181704 / f0cfb63b07df8096f89678522f0a4873b965c e71	no	2
Warnings:					

Information:

35	Non Patent Literature	N12.pdf	44539	no	2
			1770d727b45155fb48f1acba04c04b9e038 4e5c1		

Warnings:

Information:

36	Non Patent Literature	N13_CN_OA_19Oct11.pdf	757638	no	6
			36f71e1e0dfa4c0d3e411a8991859b6bff14 485b		

Warnings:

Information:

37	Non Patent Literature	N14.pdf	546955	no	5
			0ae535355d5afd8db7794c72a5b4d1eef4e d4839		

Warnings:

Information:

38	Foreign Reference	F13.pdf	38369	no	1
			cc867a44e039977e4cbb4f18e1a2f6e5ff631 5c1		

Warnings:

Information:

39	Fee Worksheet (SB06)	fee-info.pdf	33845	no	2
			5a05c3569d68a7dbcf746d1d7029f9bf7e12 837a		

Warnings:

Information:

	Total Files Size (in bytes):	17359766

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/381,219
Filing Date	March 20, 2003
First Named Inventor	YACH, David P.
Art Unit	2431
Examiner Name	AVERY, Jeremiah L.
Attorney Docket Number	13210-1465/KL

Total Number of Pages in This Submission | 412

ENCLOSURES *(Check all that apply)*

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [✓] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [✓] Extension of Time Request
- [ ] Express Abandonment Request
- [✓] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53

- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [✓] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD

- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [✓] Other Enclosure(s) (please Identify below):
  - Request for Continued Examination
  - Statement under 37 CFR 3.73(b)
  - IDS Letter
  - copies of foreign references and NPL documents

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.
Signature	*[signature]*
Printed name	Kendrick Lo
Date	November 11, 2011	Reg. No.	54,948

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	
Typed or printed name		Date	

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: RESEARCH IN MOTION LIMITED

Application No./Patent No.: 10/381,219   Filed/Issue Date: March 20, 2003

Titled: Software code signing system and method

RESEARCH IN MOTION LIMITED , a   Corporation

(Name of Assignee)   (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.

states that it is:

1. ☒   the assignee of the entire right, title, and interest in;

2. ☐   an assignee of less than the entire right, title, and interest in
(The extent (by percentage) of its ownership interest is _____ %); or

3. ☐   the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

A. ☒   An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 014188 , Frame 0164 , or for which a copy therefore is attached.

OR

B. ☐   A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____ , Frame_____ , or for which a copy thereof is attached.

2. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____ , Frame_____ , or for which a copy thereof is attached.

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____ , Frame_____ , or for which a copy thereof is attached.

☐   Additional documents in the chain of title are listed on a supplemental sheet(s).

☐   As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

_____   November 11, 2011
Signature   Date

Kendrick Lo, Regn No. 54,948   Agent for the Assignee

Printed or Typed Name   Title

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 11/16/2011

VROGERS   SALE #00000002   Mailroom Dt: 11/11/2011   022095   10381219
            01   FC : 1202   180.00 DA

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
10/381,219	03/20/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

OTHER THAN

SMALL ENTITY ☐ OR SMALL ENTITY

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

OTHER THAN

SMALL ENTITY OR SMALL ENTITY

AMENDMENT

	11/11/2011	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 112	Minus	** 109	= 3	X $ =		OR	X $60=	180
	Independent (37 CFR 1.16(h))	* 4	Minus	***12	= 0	X $ =		OR	X $250=	0
	☐ Application Size Fee (37 CFR 1.16(s))							OR		
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	**180**

AMENDMENT

		CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))							OR		
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/VIOLA ROGERS/

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

89441   7590   05/13/2011
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
05/13/2011	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dlpejeau@jonesday.com
portfolioprosecution@rim.com

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Office Action Summary	10/381,219	YACH ET AL.
	Examiner	Art Unit	
	JEREMIAH AVERY	2431	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 March 2011*.
2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *166-265* is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) *166-265* is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on *20 March 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
        1.☐ Certified copies of the priority documents have been received.
        2.☐ Certified copies of the priority documents have been received in Application No. _____.
        3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

I.   Claims 166-265 have been examined.

II.   Responses to Applicant's remarks have been given.

Response to Arguments

1.   The 35 U.S.C. 101 rejection of claims 216-240 is hereby withdrawn due to the

Applicant's amendment to independent claim 216.

2.   The 35 U.S.C. 112, first paragraph rejection of claims 191 and 216-240 is hereby

withdrawn

3.   In response to applicant's arguments against the references individually, one

cannot show nonobviousness by attacking references individually where the rejections

are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208

USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir.

1986).

4.   The claimed "application programming interface (API)" is disclosed within Gibbs

via column 5, lines 42-45, "an application programming interface ('API') for the

messaging server is added which provides access to the authenticated message server

services".

Specification

5.   The specification is objected to as failing to provide proper antecedent basis for

the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction

of the following is required: the "computer-readable storage medium" as found within

claims 191 and 216-240 does not possess a sufficient support or description within the

Applicant's Specification.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148

USPQ 459 (1966), that are applied for establishing a background for determining

obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 166-168, 170, 171, 173-193, 195, 196, 198-218, 220, 221, 223-243, 245,

246 and 248-265 are rejected under 35 U.S.C. 103(a) as being unpatentable over

United States Patent No. 6,795,919 to Gibbs et al., hereinafter Gibbs and further in view

of United States Patent No. 6,587,837 to Spagna et al., hereinafter Spagna.

6.   Regarding claims 166, 191, 216 and 241, Gibbs discloses a mobile device

(Figure 3, element 332 and Figure 4, element 452) containing software instructions

which when executed on the mobile device cause the mobile device to perform

operations for controlling access to an application platform, as well as a system, <u>non-</u>

transitory computer-readable storage medium and a method for controlling access to an

application platform, comprising:

storing a plurality of application programming interfaces (APIs), wherein each API can

be used to allow certain software applications to interact with the application platform,

and wherein at least one API comprises a sensitive API accessible upon verification of a

digital signature, receiving a software application requesting access to a sensitive API

(column 3, lines 10-18, "an authenticated message server functionally comprises a

digital service engine 120", column 5, lines 29-51, "Authenticated message server 316

can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and

61-66, column 10, lines 31-67, "electronic voting or polling system. In such a system,

the unique digital signature 132 is allocated by the authenticated message server 428 at

step 704" and column 11, lines 1-7);

using an application execution manager to:

determine whether the software application is signed, wherein a signed software

application includes a digital signature and a corresponding signature identification

(column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67,

"electronic voting or polling system. In such a system, the unique digital signature 132

is allocated by the authenticated message server 428 at step 704" and column 11, lines

1-7);

based upon a determination that the software application is signed, determine

whether the signature identifier of the sensitive API corresponds to the signature

identification of the signed software application (column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming

unique digital signature").

7.   With regards to claims 166, 191, 216 and 241, Gibbs discloses the claimed

features of an API and the interactions of software on a mobile device, as cited above.

However, Gibbs does not disclose the claimed features pertaining to a "public key".

Spagna discloses said features, as cited below.

8.   Regarding claims 166, 191, 216 and 241, Spagna discloses <u>wherein the</u>

<u>sensitive API is associated with a signature identifier and a public key</u> (column 16, lines

46-53, column 17, lines 2-14 and 38-46);

based upon a determination that the signature identifier of the sensitive API

corresponds to the signature identification of the signed software application, use the

public key associated with the sensitive API to verify authenticity of the digital signature

of the signed software application (column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the

signature itself using the Public Key 661 of the signing entity included if signed");

and upon verifying the authenticity of the digital signature, using the sensitive API to

allow the signed software application to interact with the application platform (column

46, lines 59-67 and column 47, lines 1-13).

9.   The motivation to combine would be that "the issuer of SC(s) protects the

integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

10.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to incorporate the teachings of Spagna with the teachings

of Gibbs so that "Content encryption keys are used by End-User Device(s) 109 to

unlock Content 113 for which they have obtained rights, typically by a purchase

transaction from and authorized Electronic Digital Content Store(s) 103." (*Spagna* –

column 45, lines 13-17).

11.   Regarding claims 167, 192, 217 and 242, Gibbs discloses wherein based upon a

determination that the software application requesting access to the sensitive API does

not include a signature identification, the software is not executed (Figure 6, column 8,

lines 56-65 and column 10, lines 14-30).

12.   Regarding claims 168, 193, 218 and 243, Gibbs discloses wherein based upon a

determination that the software application requesting access to the sensitive API does

not include a signature identification, the software application is denied access to the

sensitive API (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

13.   Regarding claims 170, 195, 220 and 245, Gibbs discloses wherein based upon a

determination that the digital signature is not authenticated, the software application

requesting access to the sensitive API is not executed (Figure 6, column 8, lines 56-65

and column 10, lines 14-30).

14.   Regarding claims 171, 196, 221 and 246, Gibbs discloses wherein based upon a

determination that the digital signature is not authenticated, the software application

requesting access to the sensitive API is denied access to the sensitive API (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

15.   Regarding claims 173, 198, 223 and 248, Gibbs discloses wherein a global

signature is associated with each of the plurality of APIs; and wherein the global

signature is verified prior to allowing the signed software application to interact with the application platform (column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is added which provides access to the authenticated message server services", column 8, lines 56-65 and column 10, lines 14-30).

16.   Regarding claims 174, 199, 224 and 249, Gibbs discloses wherein the application execution manager is implemented by a virtual machine (VM) (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

17.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not disclose the claim language pertaining to the "public key" and "private key" as found within the dependent claims 175, 176, 200, 201, 225, 226, 250 and 251. Spagna discloses this claim language, as cited below.

18.   Regarding claims 175, 200, 225 and 250, Spagna discloses wherein the digital signature is generated by applying a private key to a first hash of the software application; and the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same (column 16, lines 46-53, column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

19.   The motivation to combine would be that "the issuer of SC(s) protects the integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

20.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Spagna with the teachings of Gibbs so that "it is easy to calculate the digest for an input message, but it is very difficult (computationally infeasible) to generate the input message from its digest" (*Spagna* – column 17, lines 25-31).

21.   Regarding claims 176, 201, 226 and 251, Spagna discloses wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14, 25-31 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

22.   The motivation to combine would be that "the issuer of SC(s) protects the integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

23.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to incorporate the teachings of Spagna with the teachings

of Gibbs so that "it is easy to calculate the digest for an input message, but it is very

difficult (computationally infeasible) to generate the input message from its digest"

(*Spagna* – column 17, lines 25-31).

24.   Regarding claims 177, 202, 227 and 252, Gibbs discloses wherein the digital

signature is generated by a code signing authority and included with the software

application (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines

45-51 and column 8, lines 40-47).

25.   Regarding claims 178, 202, 228 and 253, Gibbs discloses wherein the operations

for controlling access to the application platform further comprise:

      displaying a description string when the software application attempts to access

the sensitive API (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still

type the unique digital signature without special software (or hardware)" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML

files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

26.   Regarding claims 179, 203, 229 and 254, Gibbs discloses wherein the

application platform comprises an operating system (column 3, lines 10-18, "an

authenticated message server functionally comprises a digital service engine 120",

column 5, lines 29-51, "Authenticated message server 316 can run on a standard

personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines

54-58, "user's personal computer" and column 10, lines 49-62).

> [As it is known, a "user's personal computer" such as the "laptop"
>
> disclosed within Gibbs would contain an operating system so that a user
>
> could utilize the particular device.]

27.   Regarding claims 180, 204, 230 and 255, Gibbs discloses wherein the

application platform is on the mobile device, and wherein the application platform

includes mobile device hardware (column 6, lines 45-60, "laptop", column 8, lines 48-55

and column 10, lines 8-13).

28.   Regarding claims 181, 205, 231 and 256, Gibbs discloses wherein the

application platform comprises a cryptographic module (column 8, lines 11-32, "MD5

function" and "SHA-1 hash function").

29.   Regarding claims 182, 206, 232 and 257, Gibbs discloses wherein the

application platform comprises a data store (column 3, lines 10-18, "an authenticated

message server functionally comprises a digital service engine 120", column 5, lines 29-

51, "Authenticated message server 316 can run on a standard personal computer" and

lines 60-65, column 6, lines 30-39 and 61-66 and column 45, lines 3-21).

30.   Regarding claims 183, 207, 233 and 258, Gibbs discloses wherein the

application platform comprises a proprietary data model (column 10, lines 31-67,

"electronic voting or polling system" and column 11, lines 1-12).

31.   Regarding claims 184, 208, 234 and 259, Gibbs discloses wherein the

application platform comprises an input and output controller (column 3, lines 49-58,

"the input to the one-way hash function" and column 8, lines 11-39, "MD5 function" and

"SHA-1 hash function").

32.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not

disclose the features within claims 185, 209, 235 and 260 pertaining to "an audit trail".

Spagna discloses said "audit trail" as cited below.

33.   Regarding claims 185, 209, 235 and 260, Spagna discloses wherein the digital

signature provides an audit trail identifying a developer of the software application

requesting access to the sensitive API (column 14, lines 20-34, "obtain an audit trail of

electronic delivery to their customers").

34.   The motivation to combine would be to have a record of "all transactions that

relate to the sale and/or permitted use of the Content 113 encrypted in a SC" (*Spagna* –

column 14, lines 2-5).

35.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to incorporate the teachings of Spagna with the teachings

of Gibbs to detect if "information in a SC has been compromised or does not comply

with the Content usage conditions" (*Spagna* – column 14, lines 27-32).

36.   Regarding claims 186, 210, 236 and 261, Gibbs discloses and wherein the digital

signature associated with the problematic software application is revocable (column 10,

lines 63-67, "One response is to completely disregard the failed unique digital signature

132 and voting responses and delete them from the unique digital signature system

400").

37.   With regards to claims 186, 210, 236 and 261, though Gibbs discloses the

revoking of a digital signature within said claims, Gibbs does not disclose the claimed

feature of "wherein a problematic software application is identified using the audit trail".

Spagna discloses said feature, as cited below.

38.   Regarding claims 186, 210, 236 and 261, Spagna discloses wherein a

problematic software application is identified using the audit trail (column 14, lines 20-

34, "obtain an audit trail of electronic delivery to their customers").

39.   The motivation and obviousness to combine for claims 186, 210, 236 and 261

are the same as applied to claims 185, 209, 235 and 260.

40.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not

disclose a "revocation list" as found within claims 187, 211, 237 and 262. Spagna

discloses said "revocation list", as cited below.

41.   Regarding claims 187, 211, 237 and 262, Spagna discloses wherein the digital

signature associated with the problematic software application is revoked, and wherein

the revoked digital signature is added to a signature revocation list (column 40, lines 24-

38).

42.   The motivation to combine would be "to insure that the SC(s) is in fact valid and

the data it contains has not been corrupted in any way" (Spagna – column 46, lines 36-

38).

43.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to incorporate the teachings of Spagna with the teachings

of Gibbs to detect if "information in a SC has been compromised or does not comply

with the Content usage conditions" (*Spagna* – column 14, lines 27-32).

44.   Regarding claims 188, 212, 238 and 263, Gibbs discloses wherein the

authenticity of the digital signature is first verified each time the software application

requesting access to the sensitive API is allowed to interact with the application platform

(Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

45.   Regarding claims 189, 213, 239 and 264, Gibbs discloses wherein the digital

signature and the signature identification correspond to a mobile device type (column 6,

lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

46.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not

disclose the "public key repository" claimed within dependent claims 190, 214, 240 and

265. Spagna discloses said "public key repository", as cited below.

47.   Regarding claims 190, 214, 240 and 265, Spagna discloses wherein associating

the sensitive API with the public key includes obtaining the public key from a public key

repository (column 45, lines 3-21, "Clearinghouse(s) 105 is responsible for the rights

management functions of the Secure Digital Content Electronic Distribution System 100.

Clearinghouse(s) 105 functions include…distribution of Content encryption keys").

48.   The motivation to combine would be to have storage means for necessary keys

for subsequent usage upon request.

49.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made to incorporate the teachings of Spagna with the teachings

of Gibbs so that "Content encryption keys are used by End-User Device(s) 109 to

unlock Content 113 for which they have obtained rights, typically by a purchase

transaction from and authorized Electronic Digital Content Store(s) 103." (*Spagna* –

column 45, lines 13-17).

50.   Claims 169, 172, 194, 197, 219, 222, 244 and 247 are rejected under 35

U.S.C. 103(a) as being unpatentable over Gibbs and Spagna as applied to independent

claims 166, 191, 216 and 241 above, and further in view of United States Patent No.

6,697,948 to Rabin et al., hereinafter Rabin.

51.   Gibbs and Spagna disclose the claimed features of the independent claims, as

cited above. However, they do not disclose the claimed features within claims 169, 172,

194, 197, 219, 222, 244 and 247 pertaining to the purging of "the software application

from the mobile device. Rabin discloses said features, as cited below.

52.   Regarding claims 169, 194, 219 and 244, Rabin discloses wherein the

application platform is on the mobile device, and wherein based upon a determination

that the software application requesting access to the sensitive API does not include a

signature identification, the application execution manager purges the software

application from the mobile device (column 16, lines 17-32, "software vendor transfers

the infringing instance of software to a guardian center so that usage supervision can be

implemented to detect attempted uses of the infringing instance of software" and

column 42, lines 13-23, "if use is to be denied, this condition is indicated by the term

'GC_DISABLED'. 'INSTALLED' followed by 'REMOVED' status terms indicate that a tag

TAG_INST_SWn for an instance of software 111-114 was formerly installed on the user

device 104 but is no longer installed and consequently is not usable.").

53.   The motivation to combine would be to provide "an infringing software detection mechanism that detects an infringing instance of software that is infringing intellectual property rights" (*Rabin* – column 16, lines 26-32).

54.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Rabin within the teachings of Gibbs and Spagna to ensure that only authorized software is being utilized by the devices upon which said software has been installed; otherwise "punitive action is taken by the supervising program in the device" (*Rabin* - column 16, lines 47-49).

55.   Regarding claims 172, 197, 222 and 247, Rabin discloses wherein the application platform is on the mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device (column 16, lines 17-32, "software vendor transfers the infringing instance of software to a guardian center so that usage supervision can be implemented to detect attempted uses of the infringing instance of software" and column 42, lines 13-23, "if use is to be denied, this condition is indicated by the term 'GC_DISABLED'. 'INSTALLED' followed by 'REMOVED' status terms indicate that a tag TAG_INST_SWn for an instance of software 111-114 was formerly installed on the user device 104 but is no longer installed and consequently is not usable.").

56.   The claim language within claims 172, 197, 222 and 247 is analogous to the claim language set forth within claims 169, 194, 219 and 244; thus the motivation and

obviousness to combine for claims 169, 194, 219 and 244 also pertain to claims 172, 197, 222 and 247.

Conclusion

57. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

58. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

59. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

60. The following United States Patents are cited to further show the state of the art with regards to software interactions on a device and the protection of the software and device, such as:

United States Patent No. 6,574,609 to Downs et al., which is cited to show a secure electronic content management system.

United States Patent No. 6,324,650 to Ogilvie which is cited to show message content protection and conditional disclosure.

United States Patent No. 6,795,923 to Stern et al., which is cited to show a

mechanism for embedding network based control systems in a local network

interface device.

United States Patent No. 6,233,683 to Chan et al., which is cited to show a

system and method for a multi-application smart card which can facilitate a post-

issuance download of an application onto the smart card.

United States Patent No. 6,390,374 to Carper et al., which is cited to show a

system and method for installing/de-installing an application on a smart card.

United States Patent No. 6,374,357 to Mohammed et al., which is cited to show a

system and method for regulating a network service provider's ability to host

distributed applications in a distributed processing environment.

United States Patent No. 6,345,256 to Milsted et al., which is cited to show an

automated method and apparatus to package digital content for electronic

distribution using the identity of the source content.

United States Patent No. 6,212,636 to Boyle et al., which is cited to show a

method for establishing trust in a computer network via association.

United States Patent No. 6,748,541 to Margalit et al., which is cited to show a

user-computer interaction method for use by a population of flexibly connectable

computer systems.

61.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to JEREMIAH AVERY whose telephone number is

(571)272-8627. The examiner can normally be reached on Monday thru Friday 8:30am-5pm.

62.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan Flynn can be reached on (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

63.   Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jeremiah Avery/
Examiner, Art Unit 2431
/NATHAN FLYNN/
Supervisory Patent Examiner, Art Unit 2468

		Application/Control No. 10/381,219	Applicant(s)/Patent Under Reexamination YACH ET AL.	
	Notice of References Cited	Examiner JEREMIAH AVERY	Art Unit 2431	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,795,919	09-2004	Gibbs et al.	713/170
*	B	US-6,587,837	07-2003	Spagna et al.	705/52
*	C	US-6,697,948	02-2004	Rabin et al.	726/30
*	D	US-6,574,609	06-2003	Downs et al.	705/50
*	E	US-6,324,650	11-2001	Ogilvie, John W.L.	726/2
*	F	US-6,795,923	09-2004	Stern et al.	726/12
*	G	US-6,233,683	05-2001	Chan et al.	713/172
*	H	US-6,390,374	05-2002	Carper et al.	235/492
*	I	US-6,374,357	04-2002	Mohammed et al.	726/5
*	J	US-6,345,256	02-2002	Milsted et al.	705/64
*	K	US-6,697,948	02-2004	Rabin et al.	726/30
*	L	US-6,212,636	04-2001	Boyle et al.	713/168
*	M	US-6,748,541	06-2004	Margalit et al.	726/9

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims		10381219	YACH ET AL.
		Examiner	**Art Unit**
		JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE									
Final	Original	05/05/2011									
	1	-									
	2	-									
	3	-									
	4	-									
	5	-									
	6	-									
	7	-									
	8	-									
	9	-									
	10	-									
	11	-									
	12	-									
	13	-									
	14	-									
	15	-									
	16	-									
	17	-									
	18	-									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
	26	-									
	27	-									
	28	-									
	29	-									
	30	-									
	31	-									
	32	-									
	33	-									
	34	-									
	35	-									
	36	-									

Index of Claims	Application/Control No. 10381219	Applicant(s)/Patent Under Reexamination YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE									
Final	Original	05/05/2011									
	37	-									
	38	-									
	39	-									
	40	-									
	41	-									
	42	-									
	43	-									
	44	-									
	45	-									
	46	-									
	47	-									
	48	-									
	49	-									
	50	-									
	51	-									
	52	-									
	53	-									
	54	-									
	55	-									
	56	-									
	57	-									
	58	-									
	59	-									
	60	-									
	61	-									
	62	-									
	63	-									
	64	-									
	65	-									
	66	-									
	67	-									
	68	-									
	69	-									
	70	-									
	71	-									
	72	-									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE
Final	Original	05/05/2011
	73	-
	74	-
	75	-
	76	-
	77	-
	78	-
	79	-
	80	-
	81	-
	82	-
	83	-
	84	-
	85	-
	86	-
	87	-
	88	-
	89	-
	90	-
	91	-
	92	-
	93	-
	94	-
	95	-
	96	-
	97	-
	98	-
	99	-
	100	-
	101	-
	102	-
	103	-
	104	-
	105	-
	106	-
	107	-
	108	-

	Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
		10381219	YACH ET AL.
		Examiner	Art Unit
		JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE									
Final	Original	05/05/2011									
	109	-									
	110	-									
	111	-									
	112	-									
	113	-									
	114	-									
	115	-									
	116	-									
	117	-									
	118	-									
	119	-									
	120	-									
	121	-									
	122	-									
	123	-									
	124	-									
	125	-									
	126	-									
	127	-									
	128	-									
	129	-									
	130	-									
	131	-									
	132	-									
	133	-									
	134	-									
	135	-									
	136	-									
	137	-									
	138	-									
	139	-									
	140	-									
	141	-									
	142	-									
	143	-									
	144	-									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	05/05/2011								
	145	-								
	146	-								
	147	-								
	148	-								
	149	-								
	150	-								
	151	-								
	152	-								
	153	-								
	154	-								
	155	-								
	156	-								
	157	-								
	158	-								
	159	-								
	160	-								
	161	-								
	162	-								
	163	-								
	164	-								
	165	-								
	166	✓								
	167	✓								
	168	✓								
	169	✓								
	170	✓								
	171	✓								
	172	✓								
	173	✓								
	174	✓								
	175	✓								
	176	✓								
	177	✓								
	178	✓								
	179	✓								
	180	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE							
Final	Original	05/05/2011							
	181	✓							
	182	✓							
	183	✓							
	184	✓							
	185	✓							
	186	✓							
	187	✓							
	188	✓							
	189	✓							
	190	✓							
	191	✓							
	192	✓							
	193	✓							
	194	✓							
	195	✓							
	196	✓							
	197	✓							
	198	✓							
	199	✓							
	200	✓							
	201	✓							
	202	✓							
	203	✓							
	204	✓							
	205	✓							
	206	✓							
	207	✓							
	208	✓							
	209	✓							
	210	✓							
	211	✓							
	212	✓							
	213	✓							
	214	✓							
	215	✓							
	216	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE								
Final	Original	05/05/2011								
	217	✓								
	218	✓								
	219	✓								
	220	✓								
	221	✓								
	222	✓								
	223	✓								
	224	✓								
	225	✓								
	226	✓								
	227	✓								
	228	✓								
	229	✓								
	230	✓								
	231	✓								
	232	✓								
	233	✓								
	234	✓								
	235	✓								
	236	✓								
	237	✓								
	238	✓								
	239	✓								
	240	✓								
	241	✓								
	242	✓								
	243	✓								
	244	✓								
	245	✓								
	246	✓								
	247	✓								
	248	✓								
	249	✓								
	250	✓								
	251	✓								
	252	✓								

U.S. Patent and Trademark Office   Part of Paper No. : 20110505

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE								
Final	Original	05/05/2011								
	253	✓								
	254	✓								
	255	✓								
	256	✓								
	257	✓								
	258	✓								
	259	✓								
	260	✓								
	261	✓								
	262	✓								
	263	✓								
	264	✓								
	265	✓								

Page 686 of 1415

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

SEARCHED

Class	Subclass	Date	Examiner
none	none	5/6/2011	JLA

SEARCH NOTES

Search Notes	Date	Examiner
Updated EAST Search	5/6/2011	JLA
UpdatedKeyword Search within Class 711, subclass 100, Class 713, subclasses 1, 176, 187 and 189. Class 395, subclass 682 and Class 719, subclass 328	5/6/2011	JLA

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
none	none	5/6/2011	JLA

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1100	(@ad<"20000921" @pd<"20000921" @rlad<"20000921") and ((API or (Application near program $4 near interface)) same (signature or key)) and (authentic$ or verify$ or verification) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant))	US-PGPUB; USPAT; EPO	OR	ON	2011/05/06 11:25
L2	106	(719/328.ccls. or 711/100.ccls. or 713/1.ccls. or 713/176.ccls. or 713/187.ccls. or 713/189.ccls. or 395/682.cxr.) and l1	US-PGPUB; USPAT; EPO	OR	ON	2011/05/06 11:27
L3	78	l2 and ((authentic$ or verify$ or verification) with signature)	US-PGPUB; USPAT; EPO	OR	ON	2011/05/06 11:28
S1	8	(digital near signature) and (authentic$ or verify$ or verificat$) and (virtual near machine) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:23
S2	8	S1 and (portab$ or mobile or handheld or laptop or pda or cell or cellular)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S3	4	S2 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (1 of 13)5/6/2011 11:54:17 AM

Page 688 of 1415

S4	35	(digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:28
S5	737	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33
S6	41	S5 and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5 or revok$ or revocat$) same (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33
S7	30	S6 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:37
S8	30	S7 and access$	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S9	30	S8 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S10	2	S9 and ((secure near hash near algorithm) or SHA?1)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:39
S11	1	S10 and public and private	US-PGPUB; USPAT	OR	ON	2009/02/20 11:40
S12	31	S6 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:46

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (2 of 13)5/6/2011 11:54:17 AM

Page 689 of 1415

S13	31	S12 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:47
S14	31	S13 and ((public or private) same key)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:48
S15	0	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) same (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53
S16	30	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) and (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53
S17	28	S16 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:07
S18	118	S5 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S19	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S20	56	S19 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:26
S21	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27

S22	56	S21 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S23	40	S22 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:28
S24	55	S22 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:32
S25	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S26	119	S25 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S27	62	S26 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S28	16	S27 and (SIM or (subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S29	30	S25 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda) and (SIM or (Subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:14
S30	16	S29 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:24
S31	9	S28 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:28

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (4 of 13)5/6/2011 11:54:17 AM

Page 691 of 1415

S32	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S33	119	S32 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S34	62	S33 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S35	57	S34 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S36	56	S35 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S37	56	S36 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S38	36	S37 and (digital near signature) and ((authentic$ or verify$ or verification) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S39	0	S38 and (signature near (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S40	3	S38 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:53
S41	40	S37 and (digital near signature) and ((authentic$ or verify$ or verification or valid$) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57
S42	6	S41 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57

S43	757	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S44	130	S43 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S45	72	S44 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S46	65	S45 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S47	64	S46 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S48	64	S47 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S49	48	S48 and (digital near signature) and ((authentic$ or verify$ or verification or valid $) near signature)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S50	6	S49 and (signature same (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S51	11	S49 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat $) near (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:37
S52	21	S49 and (virtual near machine)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:38

S53	25263	(@ad<"20000921" @prad<"20000921") and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) near (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:13
S54	31	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial or unauthentic$ or unverif$4) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:15
S55	0	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial or unauthentic$ or unverif$4) near (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:53
S56	62	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:53
S57	31	S56 and ((deny$ or denies or denial or unauthentic$ or unverif$4) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:55

S58	94	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (software or application or program)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:48
S59	14	S58 and (virtual near machine)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:50
S60	61	S58 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:51
S61	61	S60 and ($2crypt$ or $2cipher $)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:51
S62	61	S61 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:52
S63	1	S62 and S59	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:52
S64	13286	711/100.ccls. or 713/1.ccls. or 713/176.ccls. or 713/187.ccls. or 713/189.ccls. or 395/682. cxr. and (API or (application near programming near interface)) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:04

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (8 of 13)5/6/2011 11:54:17 AM

Page 695 of 1415

S65	4049	S64 and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:04
S66	2341	S65 and ((digital near signature) near$4 (authentic$ or verify$ or verificat$))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:06
S67	343	(719/328.ccls. or 711/100.ccls. or 713/1.ccls. or 713/176.ccls. or 713/187.ccls. or 713/189.ccls. or 395/682.cxr.) and (API or (application near programming near interface)) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S68	343	S67 and ((digital near signature) near$4 (authentic$ or verify$ or verificat$))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S69	115	S68 and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S70	115	S69 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S71	85	S70 and ((public and private) near key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:08
S72	94	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$5 near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) same (software or application or program)) and (@ad<"20000921"	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:11

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (9 of 13)5/6/2011 11:54:17 AM

Page 696 of 1415

		@pd<"20000921" @rlad<"20000921")				
S73	76	S71 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:12
S74	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) near$ (software or application or program)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:28
S75	1	S74 and ((hash$ or (one?way or (one near way)) or abridg$) near key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:30
S76	0	S70 and (((deny or denying or denial or restrict$ or prohibit$) near access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:37
S77	53	S70 and (((deny or denying or denial or restrict$ or prohibit$) near$ access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:37
S78	53	S70 and (((deny or denying or denied or denial or restrict$ or prohibit$) near$ access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:38
S79	53	S78 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:40

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (10 of 13)5/6/2011 11:54:17 AM

Page 697 of 1415

S80	43	S79 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:40
S81	53	S78 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:54
S82	97	S74 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:55
S83	773	(YACH-DAVID-P.in. or BROWN-MICHAEL-S.in. or LITTLE-HERBERT-A.in.)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:21
S84	180	(digital near signature) and (authentic$ or verify$ or verificat$) and S83	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:22
S85	177	S84 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:22
S86	41	S84 and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$5 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:24
S87	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) near$ (software or application or program)) and	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:16

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (11 of 13)5/6/2011 11:54:17 AM

Page 698 of 1415

		(@ad<"20000921" @pd<"20000921" @rlad<"20000921")				
S88	97	S87 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:16
S89	97	S87 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:18
S90	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:29
S91	98	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$4 near interface)) and ((deny$ or denies or denial) same (digital near signature)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:30
S92	98	S91 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:30

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (12 of 13)5/6/2011 11:54:17 AM

Page 699 of 1415

S93	101328	(@ad<"20000921" @pd<"20000921" @rlad<"20000921") and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan)) and (authentic$ or verify$ or verification) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:36
S94	32	S93 and ((digital near signature) same ((unauthentic$ or unverify$ or unverifi$) or ("not" near (authentic$ or verify$ or verification))))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:40
S95	992	S93 and ((unauthentic$ or unverify$ or unverifi$) or ("not" near (authentic$ or verify$ or verification)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:47
S96	484	S95 and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:48
S97	481	S96 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:49
S98	126	S97 and (API or (application near program$4 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:50

5/6/2011 11:54:09 AM
C:\Documents and Settings\javery\My Documents\EAST\Workspaces\10381219.wsp

file:///Cl/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (13 of 13)5/6/2011 11:54:17 AM

Page 700 of 1415

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	David P. Yach; Michael S. Brown; Herbert A. Little
Internat'l. Appl'n. No.	:	PCT/CA01/01344
Internat'l. Filing Date	:	09/20/2001
U.S. Serial No.	:	10/381,219
U.S. Filing Date	:	03/20/2003
Title	:	Software Code Signing System And Method
Art Unit	:	2431
Examiner	:	J. Avery
Docket No.	:	555255-012423

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSIVE AMENDMENT

Sir:

In response to the non-final Office Action dated September 2, 2010, please amend the above-identified application as follows and consider the remarks herein.

CLI-1879695v1

Amendments to the claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-165. (Cancelled).

166.   (Currently Amended) A mobile device containing software instructions which when executed on the mobile device cause the mobile device to perform operations for controlling access to an application platform, the operations comprising:

   storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;

   receiving a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

   using an application execution manager to:

      determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

      based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

      based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

   upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

167.　(Previously Presented)　　　The mobile device of claim 166, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

168.　(Previously Presented)　　　The mobile device of claim 166, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

169.　(Previously Presented)　　　The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

170.　(Previously Presented)　　　The mobile device of claim 166, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

171.　(Previously Presented)　　　The mobile device of claim 166, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

172.　(Previously Presented)　　　The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

173.　(Previously Presented)　　　The mobile device of claim 166, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

174.   (Previously Presented)   The mobile device of claim 166, wherein the application execution manager is implemented by a virtual machine (VM).

175.   (Previously Presented)   The mobile device of claim 166, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

176.   (Previously Presented)   The mobile device of claim 166, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

177.   (Previously Presented)   The mobile device of claim 166, wherein the digital signature is generated by a code signing authority and included with the software application.

178.   (Previously Presented)   The mobile device of claim 166, wherein the operations for controlling access to the application platform further comprise:

displaying a description string when the software application attempts to access the sensitive API.

179.   (Previously Presented)   The mobile device of claim 166, wherein the application platform comprises an operating system.

180.   (Previously Presented)   The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein the application platform includes mobile device hardware.

181.   (Previously Presented)   The mobile device of claim 166, wherein the application platform comprises a cryptographic module.

182.   (Previously Presented)   The mobile device of claim 166, wherein the application platform comprises a data store.

183.   (Previously Presented)   The mobile device of claim 166, wherein the application platform comprises a proprietary data model.

184.   (Previously Presented)   The mobile device of claim 166, wherein the application platform comprises an input and output controller.

185.   (Previously Presented)   The mobile device of claim 166, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

186.   (Previously Presented)   The mobile device of claim 185, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

187.   (Previously Presented)   The mobile device of claim 186, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

188.   (Previously Presented)   The mobile device of claim 166, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

189.   (Previously Presented)   The mobile device of claim 166, wherein the digital signature and the signature identification correspond to a mobile device type.

190.   (Previously Presented)   The mobile device of claim 166, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

191.   (Currently Amended) A system for controlling access to an application platform, comprising:

one or more processors;

one or more computer-readable storage mediums containing software instructions executable on the one or more processors to cause the one or more processors to perform operations including:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;

receiving a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

192. (Previously Presented)   The system of claim 191, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

193. (Previously Presented)   The system of claim 191, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

194. (Previously Presented)   The system of claim 191, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

195. (Previously Presented)   The system of claim 191, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

196. (Previously Presented)   The system of claim 191, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

197. (Previously Presented)   The system of claim 191, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

198. (Previously Presented)   The system of claim 191, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

199. (Previously Presented)   The system of claim 191, wherein the application execution manager is implemented by a virtual machine (VM).

200. (Previously Presented)   The system of claim 191, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

201. (Previously Presented)   The system of claim 191, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

202. (Previously Presented)   The system of claim 191, further comprising:

a code signing authority, wherein the code signing authority determines whether the software application should be given access to a sensitive API, and based upon a determination that the software application should be given access to a sensitive API, the code signing authority accepts the software application and generates a digital signature that is included with the software application.

203. (Previously Presented)   The system of claim 191, wherein the operations performed by the one or more processors further comprise:

displaying a description string when the software application attempts to access the sensitive API.

204. (Previously Presented)   The system of claim 191, wherein the application platform comprises an operating system.

205.   (Previously Presented)   The system of claim 191, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

206.   (Previously Presented)   The system of claim 191, wherein the application platform comprises a cryptographic module.

207.   (Previously Presented)   The system of claim 191, wherein the application platform comprises a data store.

208.   (Previously Presented)   The system of claim 191, wherein the application platform comprises a proprietary data model.

209.   (Previously Presented)   The system of claim 191, wherein the application platform comprises an input and output controller.

210.   (Previously Presented)   The system of claim 191, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

211.   (Previously Presented)   The system of claim 210, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

212.   (Previously Presented)   The system of claim 211, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

213.   (Previously Presented)   The system of claim 191, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

214. (Previously Presented)   The system of claim 191, wherein the digital signature and the signature identification correspond to a mobile device type.

215. (Previously Presented)   The system of claim 191, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

216. (Currently Amended) A <u>non-transitory</u> computer-readable storage medium encoded with instructions that when executed on one or more processors within a computer system perform a method for controlling access to an application platform, the method comprising:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature<u>, wherein the sensitive API is associated with a signature identifier and a public key</u>;

receiving a software application requesting access to a sensitive API~~, wherein the sensitive API is associated with a signature identifier and a public key~~;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

217. (Previously Presented)   The computer-readable storage medium of claim 216, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

CLI-1879695v1

218. (Previously Presented)   The computer-readable storage medium of claim 216, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

219. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

220. (Previously Presented)   The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

221. (Previously Presented)   The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

222. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

223. (Previously Presented)   The computer-readable storage medium of claim 216, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

224. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application execution manager is implemented by a virtual machine (VM).

225.   (Previously Presented)   The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

226.   (Previously Presented)   The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

227.   (Previously Presented)   The computer-readable storage medium of claim 216, wherein the digital signature is generated by a code signing authority and included with the software application.

228.   (Previously Presented)   The computer-readable storage medium of claim 216, further comprising:

displaying a description string when the software application attempts to access the sensitive API.

229.   (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform comprises an operating system.

230.   (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

Page 712 of 1415

231. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform comprises a cryptographic module.

232. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform comprises a data store.

233. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform comprises a proprietary data model.

234. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the application platform comprises an input and output controller.

235. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

236. (Previously Presented)   The computer-readable storage medium of claim 235, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

237. (Previously Presented)   The computer-readable storage medium of claim 236, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

238. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

239. (Previously Presented)   The computer-readable storage medium of claim 216, wherein the digital signature and the signature identification correspond to a mobile device type.

240. (Previously Presented)   The computer-readable storage medium of claim 216, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

241. (Currently Amended) A method for controlling access to an application platform, comprising:

    storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;

    receiving, using one or more processors, a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

    using an application execution manager to:

        determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

        based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

        based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

        upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

CLI-1879695v1

Page 714 of 1415

242.   (Previously Presented)   The method of claim 241, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

243.   (Previously Presented)   The method of claim 241, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

244.   (Previously Presented)   The method of claim 241, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

245.   (Previously Presented)   The method of claim 241, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

246.   (Previously Presented)   The mobile device of claim 241, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

247.   (Previously Presented)   The method of claim 241, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

248.   (Previously Presented)   The method of claim 241, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

249. (Previously Presented) The method of claim 241, wherein the application execution manager is implemented by a virtual machine (VM).

250. (Previously Presented) The method of claim 241, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

251. (Previously Presented) The method of claim 241, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

252. (Previously Presented) The method of claim 241, further comprising:

determining by a code signing authority, whether the software application should be given access to a sensitive API, wherein based upon a determination that the software application should be given access to a sensitive API, the code signing authority accepts the software application and generates a digital signature that is included with the software application.

253. (Previously Presented) The method of claim 241, wherein the operations for controlling access to the application platform further comprise:

displaying a description string when the software application attempts to access the sensitive API.

254. (Previously Presented) The method of claim 241, wherein the application platform comprises an operating system.

255.   (Previously Presented)   The method of claim 241, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

256.   (Previously Presented)   The method of claim 241, wherein the application platform comprises a cryptographic module.

257.   (Previously Presented)   The method of claim 241, wherein the application platform comprises a data store.

258.   (Previously Presented)   The method of claim 241, wherein the application platform comprises a proprietary data model.

259.   (Previously Presented)   The method of claim 241, wherein the application platform comprises an input and output controller.

260.   (Previously Presented)   The method of claim 241, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

261.   (Previously Presented)   The method of claim 260, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

262.   (Previously Presented)   The method of claim 261, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

263.   (Previously Presented)   The method of claim 241, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

264. (Previously Presented)   The method of claim 241, wherein the digital signature and the signature identification correspond to a mobile device type.

265. (Previously Presented)   The method of claim 241, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

REMARKS

This paper responds to the non-final Office Action dated September 2, 2010. Claims 166-265 are pending and stand rejected. Claims 166, 191, 216, and 241 are amended. The amendments have support throughout the specification, including at Fig. 4 and the corresponding description at paragraphs [0039]-[0041]. Assignee respectfully traverses the rejections. Reconsideration of the application is respectfully requested in light of the amendments and remarks herein.

Claim Rejections – 35 U.S.C. § 112

Claims 191 and 216-240 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Specifically, the Office asserts that the claimed "computer-readable storage medium" does not possess sufficient support within the specification. Support for "computer-readable storage medium" can be found throughout the specification, including at Fig. 6 in which flash memory 624 can be used to store the software and APIs as described in paragraph [0054]. It is respectfully requested that the § 112 rejections of these claims be withdrawn.

Claim Rejections – 35 U.S.C. § 101

Claims 216-240 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Independent claim 216 is amended to recite "a non-transitory computer readable storage medium." In light of the amendment, it is respectfully requested that the § 101 rejections of these claims be withdrawn.

Claim Rejections – 35 U.S.C. § 103

Claims 166-168, 170, 171, 173-193, 195, 196, 198-218, 220, 221, 223-243, 245, 246 and 248-265 are rejected under 35 U.S.C. §103(a) as allegedly unpatentable over U.S. Patent No. 6,795,919 to Gibbs et al., and further in view of U.S. Patent No. 6,587,837 to Spagna et al. Claims 169, 172, 194, 197, 219, 222, 244 and 247 are rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Gibbs and Spagna, and further in view of U.S. Patent No. 6,697,948 to Rabin et al.

CLI-1879695v1

Amended claim 166 recites that <u>the sensitive API is associated with a signature identifier</u> <u>and a public key</u>. However, Gibbs and Spagna do not disclose such a sensitive API. Only one sentence of Gibbs mentions an API:

> The interconnection or coupling mechanism between the various connectors on the devices of the unique digital signature system 300 is preferably a fiber optic network cable, but it can also be a twisted pair, or a wireless interconnection. According to one embodiment, server 308 is a Sun Microsystems SPARC^{TM} system running electronic message software such as Oracle Corporation's InterOffice^{TM} messaging server. Router 312 is a commercially available internet router such as a Cisco Systems 7500 Series router. Authenticated message server 316 can run on a standard personal computer, such as an Intel Pentium^{TM} based microprocessor system. However, authenticated message server 316 is alternatively part of the software component stack added to server 308. In such an embodiment, ***an application programming interface ("API") for the messaging server 308 is added which provides access to the authenticated message server services***. Authenticated message server services include generating and authenticating unique digital signatures as described herein. The unique digital signature system 300 can be highly distributed, wherein incoming and outgoing messages are handled by separate servers or computer systems on an interconnected network (e.g. a LAN). col. 5, lines 29-51. (emphasis added)

Gibbs merely teaches that an API can be included for the messaging server to provide access to the messaging server services. Gibbs never discloses a sensitive API that is associated with a signature identifier and a public key. Spagna does not make up for Gibbs' deficiency. The portion of Spagna cited by the examiner provides:

> To secure the transmission of a confidential message using a public key algorithm, one must use the recipient's public key to encrypt the message. Only the recipient, who has the associated private key, can decrypt the message. Public key algorithms are also used to generate digital signatures. The private key is used for that purpose. The following section provides information on digital signatures. col. 16, lines 46-53.

This portion of Spagna merely discloses using a public key algorithm for encryption and digital signature. Spagna does not disclose any sensitive API associated with a signature identifier and a public key either.

Further, claim 166 recites <u>determin[ing] whether the software application is signed,</u> <u>wherein a signed software application includes a digital signature and a corresponding signature</u> <u>identification</u>. The Office cites to Gibbs as disclosing this feature. However, Gibbs does not

CLI-1879695v1

disclose a determination of whether a software application is signed. The unique digital signature of Gibbs is associated with a message (*e.g.*, Gibbs at FIG. 3 and col. 5, lines 45-51), a user (*e.g.*, an e-commerce user as described at Gibbs col. 6, lines 51-60, an electronic voter or polling system participant as described at Gibbs col. 10, lines 31-67 and col. 11, lines 1-7), and a request for a service (*e.g.*, send media to a remote user as described at Gibbs col. 7, lines 34-42). Nowhere does Gibbs disclose a signed software application, as recited in claim 166. Because Gibbs does not disclose a signed software application, it cannot disclose determining whether a software application is signed. Further, Gibbs cannot disclose the following step of: "based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application," because that step relies on a signed software application.

Additionally, claim 166 recites "A mobile device" as performing the operations of claim 166. In contrast, Gibbs discloses operations occurring within a client-server environment, such as shown in figures 3 or 4 of Gibbs. Because Claim 166 recites a mobile device as performing the operations, other entities such as the authentication server of Gibbs do not execute the operations of claim 166.

Because the cited references, singly or in combination, do not disclose the above-discussed limitations of claim 166, it is respectfully requested that the § 103 rejection of claim 166 be withdrawn.
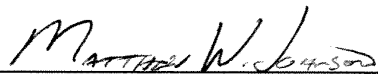
Other independent claims 191, 216, and 241 are amended to recite similar subject matter as claim 166. Thus, these independent claims are allowable for at least the reasons set forth above with respect to claim 166. In addition, it is noted that the assignee has not provided arguments with respect to certain of the dependent claims in the instant application. This is done without prejudice to the assignee's right to present arguments regarding any of the dependent claims at any point in the future. Further, because each of the dependent claims in the instant application depends from a base claim that is itself allowable, the dependent claims are allowable for at least the reasons set forth with respect to the base claims.

CONCLUSION

For the foregoing reasons, assignee respectfully submits that the pending claims are allowable. Therefore, the examiner is respectfully requested to pass this case to issuance.

Respectfully submitted,

March 2, 2011

By: _____
Matthew W. Johnson
Reg. No. 59,108
JONES DAY
North Point
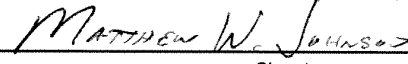901 Lakeside Avenue
Cleveland, Ohio 44114
(412) 394-9524

- 22 -

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2009** *(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number (Optional) 555255-012423
Application Number 10/381,219	Filed March 20, 2003

For   Software Code Signing System and Method

Art Unit 2431	Examiner J. Avery

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Small Entity Fee	
☐	One month (37 CFR 1.17(a)(1))	$130	$65	$_____
☐	Two months (37 CFR 1.17(a)(2))	$490	$245	$_____
☑	Three months (37 CFR 1.17(a)(3))	$1110	$555	$1110
☐	Four months (37 CFR 1.17(a)(4))	$1730	$865	$_____
☐	Five months (37 CFR 1.17(a)(5))	$2350	$1175	$_____

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-3013_____.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the   ☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☑ attorney or agent of record. Registration Number 59,108_____

☐ attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

/MATTHEW W. JOHNSON/	March 2, 2011
Signature	Date
Matthew W. Johnson	412-394-9524
Typed or printed name	Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

Electronic Acknowledgement Receipt

EFS ID:	9573338
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Stephen D. Scanlon/Matthew Johnson
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	555255012423
Receipt Date:	02-MAR-2011
Filing Date:	20-MAR-2003
Time Stamp:	18:09:28
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		012423.PDF	1504148 ddf26bd90f91d33714efdd2e326f5a31d20f2d86	yes	23

Multipart Description/PDF files in .zip description		
Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	18
Applicant Arguments/Remarks Made in an Amendment	19	22
Extension of Time	23	23

Warnings:

Information:

Total Files Size (in bytes):	1504148

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 10/381,219	Filing Date 03/20/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐	OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))						
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	

APPLICATION AS AMENDED – PART II

		(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
AMENDMENT	**03/02/2011**	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 100	Minus	** 109	= 0	X $ =		OR	X $52=	0
	Independent (37 CFR 1.16(h))	* 4	Minus	*** 12	= 0	X $ =		OR	X $220=	0
	☐ Application Size Fee (37 CFR 1.16(s))							OR		
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

		(Column 1)		(Column 2)	(Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA					
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))							OR		
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/RUTH M. LLOYD/

Document code: WFEE

RLLOYD   SALE #00000003   Mailroom Dt: 03/02/2011   503013   10381219
                01   FC : 1253   1,110.00 DA

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

89441   7590   09/02/2010
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
09/02/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dlpejeau@jonesday.com
portfolioprosecution@rim.com

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Office Action Summary	10/381,219	YACH ET AL.
	Examiner	Art Unit	
	JEREMIAH AVERY	2431	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>22 March 2010</u>.
2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>166-265</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) <u>166-265</u> is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on <u>20 March 2003</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
      1.☐ Certified copies of the priority documents have been received.
      2.☐ Certified copies of the priority documents have been received in Application No. _____.
      3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

I.   Claims 57-165 have been cancelled.

II.   Claims 166-265 have been added.

III.   Claims 166-265 have been examined.

IV.   Responses to Applicant's remarks have been given.

Continued Examination Under 37 CFR 1.114

1.   A request for continued examination under 37 CFR 1.114, including the

fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection.

Since this application is eligible for continued examination under 37 CFR 1.114,

and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the

previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Applicant's submission filed on 03/22/10 has been entered.

Response to Arguments

2.   Applicant's arguments fail to comply with 37 CFR 1.111(b) because they

amount to a general allegation that the claims define a patentable invention

without specifically pointing out how the language of the claims patentably

distinguishes them from the references. Further, new grounds of rejection are

set forth below.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and
process of making and using it, in such full, clear, concise, and exact terms as to enable any
person skilled in the art to which it pertains, or with which it is most nearly connected, to make
and use the same and shall set forth the best mode contemplated by the inventor of carrying
out his invention.

3.   Claims 191 and 216-240 are rejected under 35 U.S.C. 112, first

paragraph, as failing to comply with the written description requirement. The

claim(s) contains subject matter which was not described in the specification in

such a way as to reasonably convey to one skilled in the relevant art that the

inventor(s), at the time the application was filed, had possession of the claimed

invention. The claimed "computer-readable storage medium" does not possess

sufficient support within the Applicant's Specification. There are no examples or

definitions provided regarding what the claimed "computer-readable storage

medium" is pertaining to.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or
composition of matter, or any new and useful improvement thereof, may obtain a patent
therefor, subject to the conditions and requirements of this title.

4.   Claims 216-240 are rejected under 35 U.S.C. 101 because the claimed

invention is directed to non-statutory subject matter. The claimed "computer-

readable storage medium" is not properly defined to encompass solely statutory

subject matter. Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for

all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described
as set forth in section 102 of this title, if the differences between the subject matter sought to
be patented and the prior art are such that the subject matter as a whole would have been
obvious at the time the invention was made to a person having ordinary skill in the art to which
said subject matter pertains. Patentability shall not be negatived by the manner in which the
invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1,

148 USPQ 459 (1966), that are applied for establishing a background for

determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 166-168, 170, 171, 173-193, 195, 196, 198-218, 220, 221, 223-

243, 245, 246 and 248-265 are rejected under 35 U.S.C. 103(a) as being

unpatentable over United States Patent No. 6,795,919 to Gibbs et al., hereinafter

Gibbs and further in view of United States Patent No. 6,587,837 to Spagna et al.,

hereinafter Spagna.

5.   Regarding claims 166, 191, 216 and 241, Gibbs discloses a mobile device

(Figure 3, element 332 and Figure 4, element 452) containing software

instructions which when executed on the mobile device cause the mobile device

to perform operations for controlling access to an application platform, as well as

a system, computer-readable storage medium and a method for controlling

access to an application platform, comprising:

storing a plurality of application programming interfaces (APIs), wherein each API

can be used to allow certain software applications to interact with the application

platform, and wherein at least one API comprises a sensitive API accessible

upon verification of a digital signature (column 3, lines 10-18, "an authenticated

message server functionally comprises a digital service engine 120", column 5,

lines 29-51, "Authenticated message server 316 can run on a standard personal

computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 10, lines 31-

67, "electronic voting or polling system. In such a system, the unique digital

signature 132 is allocated by the authenticated message server 428 at step 704"

and column 11, lines 1-7);

using an application execution manager to:

    determine whether the software application is signed, wherein a signed

software application includes a digital signature and a corresponding signature

identification (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10,

lines 31-67, "electronic voting or polling system. In such a system, the unique

digital signature 132 is allocated by the authenticated message server 428 at

step 704" and column 11, lines 1-7);

    based upon a determination that the software application is signed,

determine whether the signature identifier of the sensitive API corresponds to the

signature identification of the signed software application (column 9, lines 36-58,

"an adapted digital signature 144, is compared against the adapted digital

signature in the incoming unique digital signature").

6.   With regards to claims 166, 191, 216 and 241, Gibbs discloses the

claimed features of an API and the interactions of software on a mobile device,

as cited above. However, Gibbs does not disclose the claimed features

pertaining to a "public key". Spagna discloses said features, as cited below.

7.   Regarding claims 166, 191, 216 and 241, Spagna discloses receiving a

software application requesting access to a sensitive API, wherein the sensitive

API is associated with a signature identifier and a public key (column 16, lines
46-53, column 17, lines 2-14 and 38-46);

based upon a determination that the signature identifier of the sensitive API
corresponds to the signature identification of the signed software application, use
the public key associated with the sensitive API to verify authenticity of the digital
signature of the signed software application (column 46, lines 40-58, "To validate
the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents
631 of the signature itself using the Public Key 661 of the signing entity included
if signed"); and

upon verifying the authenticity of the digital signature, using the sensitive API to
allow the signed software application to interact with the application platform
(column 46, lines 59-67 and column 47, lines 1-13).

8.   The motivation to combine would be that "the issuer of SC(s) protects the
integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

9.   Therefore, it would have been obvious to one of ordinary skill in the art at
the time the invention was made to incorporate the teachings of Spagna with the
teachings of Gibbs so that "Content encryption keys are used by End-User
Device(s) 109 to unlock Content 113 for which they have obtained rights,
typically by a purchase transaction from and authorized Electronic Digital Content
Store(s) 103." (*Spagna* – column 45, lines 13-17).

10.   Regarding claims 167, 192, 217 and 242, Gibbs discloses wherein based
upon a determination that the software application requesting access to the

sensitive API does not include a signature identification, the software is not

executed (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

11.   Regarding claims 168, 193, 218 and 243, Gibbs discloses wherein based

upon a determination that the software application requesting access to the

sensitive API does not include a signature identification, the software application

is denied access to the sensitive API (Figure 6, column 8, lines 56-65 and

column 10, lines 14-30).

12.   Regarding claims 170, 195, 220 and 245, Gibbs discloses wherein based

upon a determination that the digital signature is not authenticated, the software

application requesting access to the sensitive API is not executed (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

13.   Regarding claims 171, 196, 221 and 246, Gibbs discloses wherein based

upon a determination that the digital signature is not authenticated, the software

application requesting access to the sensitive API is denied access to the

sensitive API (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

14.   Regarding claims 173, 198, 223 and 248, Gibbs discloses wherein a

global signature is associated with each of the plurality of APIs; and wherein the

global signature is verified prior to allowing the signed software application to

interact with the application platform (column 5, lines 40-51, "an application

server interface ('API') for messaging server 308 is added which provides access

to the authenticated message server services", column 8, lines 56-65 and column

10, lines 14-30).

15.   Regarding claims 174, 199, 224 and 249, Gibbs discloses wherein the

application execution manager is implemented by a virtual machine (VM)

(column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-

30 and 35-45).

16.   Gibbs discloses the claimed invention, as cited above. However, Gibbs

does not disclose the claim language pertaining to the "public key" and "private

key" as found within the dependent claims 175, 176, 200, 201, 225, 226, 250 and

251. Spagna discloses this claim language, as cited below.

17.   Regarding claims 175, 200, 225 and 250, Spagna discloses wherein the

digital signature is generated by applying a private key to a first hash of the

software application; and the digital signature is authenticated by generating a

second hash of the software application to obtain a generated hash, applying the

public key to the digital signature to obtain a recovered hash, and verifying that

the generated hash and the recovered hash are the same (column 16, lines 46-

53, column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate

which private key was used to create the signature thus in EU is the End-User(s)

digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58,

"To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts

the Contents 631 of the signature itself using the Public Key 661 of the signing

entity included if signed").

18.   The motivation to combine would be that "the issuer of SC(s) protects the

integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

19.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Spagna with the teachings of Gibbs so that "it is easy to calculate the digest for an input message, but it is very difficult (computationally infeasible) to generate the input message from its digest" (*Spagna* – column 17, lines 25-31).

20.   Regarding claims 176, 201, 226 and 251, Spagna discloses wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14, 25-31 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

21.   The motivation to combine would be that "the issuer of SC(s) protects the integrity of SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

22.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Spagna with the

teachings of Gibbs so that "it is easy to calculate the digest for an input message, but it is very difficult (computationally infeasible) to generate the input message from its digest" (*Spagna* – column 17, lines 25-31).

23.   Regarding claims 177, 202, 227 and 252, Gibbs discloses wherein the digital signature is generated by a code signing authority and included with the software application (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

24.   Regarding claims 178, 202, 228 and 253, Gibbs discloses wherein the operations for controlling access to the application platform further comprise:

       displaying a description string when the software application attempts to access the sensitive API (column 8, lines 28-39, "a limited character ASCII set is used since remote users on legacy electronic message and existing telephone systems can still type the unique digital signature without special software (or hardware)" and column 10, lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

25.   Regarding claims 179, 203, 229 and 254, Gibbs discloses wherein the application platform comprises an operating system (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62).

[As it is known, a "user's personal computer" such as the "laptop"

disclosed within Gibbs would contain an operating system so that a

user could utilize the particular device.]

26.   Regarding claims 180, 204, 230 and 255, Gibbs discloses wherein the

application platform is on the mobile device, and wherein the application platform

includes mobile device hardware (column 6, lines 45-60, "laptop", column 8, lines

48-55 and column 10, lines 8-13).

27.   Regarding claims 181, 205, 231 and 256, Gibbs discloses wherein the

application platform comprises a cryptographic module (column 8, lines 11-32,

"MD5 function" and "SHA-1 hash function").

28.   Regarding claims 182, 206, 232 and 257, Gibbs discloses wherein the

application platform comprises a data store (column 3, lines 10-18, "an

authenticated message server functionally comprises a digital service engine

120", column 5, lines 29-51, "Authenticated message server 316 can run on a

standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66

and column 45, lines 3-21).

29.   Regarding claims 183, 207, 233 and 258, Gibbs discloses wherein the

application platform comprises a proprietary data model (column 10, lines 31-67,

"electronic voting or polling system" and column 11, lines 1-12).

30.   Regarding claims 184, 208, 234 and 259, Gibbs discloses wherein the

application platform comprises an input and output controller (column 3, lines 49-

58, "the input to the one-way hash function" and column 8, lines 11-39, "MD5

function" and "SHA-1 hash function").

31.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not disclose the features within claims 185, 209, 235 and 260 pertaining to "an audit trail". Spagna discloses said "audit trail" as cited below.

32.   Regarding claims 185, 209, 235 and 260, Spagna discloses wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API (column 14, lines 20-34, "obtain an audit trail of electronic delivery to their customers").

33.   The motivation to combine would be to have a record of "all transactions that relate to the sale and/or permitted use of the Content 113 encrypted in a SC" (*Spagna* – column 14, lines 2-5).

34.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Spagna with the teachings of Gibbs to detect if "information in a SC has been compromised or does not comply with the Content usage conditions" (*Spagna* – column 14, lines 27-32).

35.   Regarding claims 186, 210, 236 and 261, Gibbs discloses and wherein the digital signature associated with the problematic software application is revocable (column 10, lines 63-67, "One response is to completely disregard the failed unique digital signature 132 and voting responses and delete them from the unique digital signature system 400").

36.   With regards to claims 186, 210, 236 and 261, though Gibbs discloses the revoking of a digital signature within said claims, Gibbs does not disclose the

claimed feature of "wherein a problematic software application is identified using the audit trail". Spagna discloses said feature, as cited below.

37.   Regarding claims 186, 210, 236 and 261, Spagna discloses wherein a problematic software application is identified using the audit trail (column 14, lines 20-34, "obtain an audit trail of electronic delivery to their customers").

38.   The motivation and obviousness to combine for claims 186, 210, 236 and 261 are the same as applied to claims 185, 209, 235 and 260.

39.   Gibbs discloses the claimed invention, as cited above. However, Gibbs does not disclose a "revocation list" as found within claims 187, 211, 237 and 262. Spagna discloses said "revocation list", as cited below.

40.   Regarding claims 187, 211, 237 and 262, Spagna discloses wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list (column 40, lines 24-38).

41.   The motivation to combine would be "to insure that the SC(s) is in fact valid and the data it contains has not been corrupted in any way" (*Spagna* – column 46, lines 36-38).

42.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Spagna with the teachings of Gibbs to detect if "information in a SC has been compromised or does not comply with the Content usage conditions" (*Spagna* – column 14, lines 27-32).

43.   Regarding claims 188, 212, 238 and 263, Gibbs discloses wherein the

authenticity of the digital signature is first verified each time the software

application requesting access to the sensitive API is allowed to interact with the

application platform (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

44.   Regarding claims 189, 213, 239 and 264, Gibbs discloses wherein the

digital signature and the signature identification correspond to a mobile device

type (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines

8-13).

45.   Gibbs discloses the claimed invention, as cited above. However, Gibbs

does not disclose the "public key repository" claimed within dependent claims

190, 214, 240 and 265. Spagna discloses said "public key repository", as cited

below.

46.   Regarding claims 190, 214, 240 and 265, Spagna discloses wherein

associating the sensitive API with the public key includes obtaining the public key

from a public key repository (column 45, lines 3-21, "Clearinghouse(s) 105 is

responsible for the rights management functions of the Secure Digital Content

Electronic Distribution System 100. Clearinghouse(s) 105 functions

include...distribution of Content encryption keys").

47.   The motivation to combine would be to have storage means for necessary

keys for subsequent usage upon request.

48.   Therefore, it would have been obvious to one of ordinary skill in the art at

the time the invention was made to incorporate the teachings of Spagna with the

teachings of Gibbs so that "Content encryption keys are used by End-User

Device(s) 109 to unlock Content 113 for which they have obtained rights,

typically by a purchase transaction from and authorized Electronic Digital Content

Store(s) 103." (*Spagna* – column 45, lines 13-17).

49. Claims 169, 172, 194, 197, 219, 222, 244 and 247 are rejected under 35

U.S.C. 103(a) as being unpatentable over Gibbs and Spagna as applied to

independent claims 166, 191, 216 and 241 above, and further in view of United

States Patent No. 6,697,948 to Rabin et al., hereinafter Rabin.

50. Gibbs and Spagna disclose the claimed features of the independent

claims, as cited above. However, they do not disclose the claimed features

within claims 169, 172, 194, 197, 219, 222, 244 and 247 pertaining to the purging

of "the software application from the mobile device. Rabin discloses said

features, as cited below.

51. Regarding claims 169, 194, 219 and 244, Rabin discloses wherein the

application platform is on the mobile device, and wherein based upon a

determination that the software application requesting access to the sensitive API

does not include a signature identification, the application execution manager

purges the software application from the mobile device (column 16, lines 17-32,

"software vendor transfers the infringing instance of software to a guardian center

so that usage supervision can be implemented to detect attempted uses of the

infringing instance of software" and column 42, lines 13-23, "if use is to be

denied, this condition is indicated by the term 'GC_DISABLED'. 'INSTALLED'

followed by 'REMOVED' status terms indicate that a tag TAG_INST_SWn for an

instance of software 111-114 was formerly installed on the user device 104 but is no longer installed and consequently is not usable.").

52.   The motivation to combine would be to provide "an infringing software detection mechanism that detects an infringing instance of software that is infringing intellectual property rights" (*Rabin* – column 16, lines 26-32).

53.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teachings of Rabin within the teachings of Gibbs and Spagna to ensure that only authorized software is being utilized by the devices upon which said software has been installed; otherwise "punitive action is taken by the supervising program in the device" (*Rabin* - column 16, lines 47-49).

54.   Regarding claims 172, 197, 222 and 247, Rabin discloses wherein the application platform is on the mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device (column 16, lines 17-32, "software vendor transfers the infringing instance of software to a guardian center so that usage supervision can be implemented to detect attempted uses of the infringing instance of software" and column 42, lines 13-23, "if use is to be denied, this condition is indicated by the term 'GC_DISABLED'. 'INSTALLED' followed by 'REMOVED' status terms indicate that a tag TAG_INST_SWn for an instance of software 111-114 was formerly installed on the user device 104 but is no longer installed and consequently is not usable.").

55.   The claim language within claims 172, 197, 222 and 247 is analogous to

the claim language set forth within claims 169, 194, 219 and 244; thus the

motivation and obviousness to combine for claims 169, 194, 219 and 244 also

pertain to claims 172, 197, 222 and 247.

Conclusion

56.   The prior art made of record and not relied upon is considered pertinent to

applicant's disclosure.

57.   The following United States Patents are cited to further show the state of

the art with regards to software interactions on a device and the protection of the

software and device, such as:

   United States Patent No. 6,574,609 to Downs et al., which is cited to show

   a secure electronic content management system.

   United States Patent No. 6,324,650 to Ogilvie which is cited to show

   message content protection and conditional disclosure.

   United States Patent No. 6,795,923 to Stern et al., which is cited to show

   a mechanism for embedding network based control systems in a local

   network interface device.

   United States Patent No. 6,233,683 to Chan et al., which is cited to show

   a system and method for a multi-application smart card which can

   facilitate a post-issuance download of an application onto the smart card.

   United States Patent No. 6,390,374 to Carper et al., which is cited to show

   a system and method for installing/de-installing an application on a smart

   card.

United States Patent No. 6,374,357 to Mohammed et al., which is cited to

show a system and method for regulating a network service provider's

ability to host distributed applications in a distributed processing

environment.

United States Patent No. 6,345,256 to Milsted et al., which is cited to show

an automated method and apparatus to package digital content for

electronic distribution using the identity of the source content.

58.   Any inquiry concerning this communication or earlier communications from

the examiner should be directed to JEREMIAH AVERY whose telephone number

is (571)272-8627. The examiner can normally be reached on Monday thru

Friday 8:30am-5pm.

59.   If attempts to reach the examiner by telephone are unsuccessful, the

examiner's supervisor, William Korzuch can be reached on (571) 272-7589. The

fax phone number for the organization where this application or proceeding is

assigned is 571-273-8300.

60.   Information regarding the status of an application may be obtained from

the Patent Application Information Retrieval (PAIR) system. Status information

for published applications may be obtained from either Private PAIR or Public

PAIR. Status information for unpublished applications is available through

Private PAIR only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free). If you would like assistance from a USPTO Customer Service

Representative or access to the automated information system, call 800-786-

9199 (IN USA OR CANADA) or 571-272-1000.

/Jeremiah Avery/
Examiner, Art Unit 2431
/Syed   Zia/
Primary Examiner, Art Unit 2431

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Notice of References Cited		10/381,219	YACH ET AL.
		Examiner	Art Unit	
		JEREMIAH AVERY	2431	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,795,919	09-2004	Gibbs et al.	713/170
*	B	US-6,587,837	07-2003	Spagna et al.	705/26
*	C	US-6,574,609	06-2003	Downs et al.	705/50
*	D	US-6,697,948	02-2004	Rabin et al.	726/30
*	E	US-6,345,256	02-2002	Milsted et al.	705/64
*	F	US-6,374,357	04-2002	Mohammed et al.	726/5
*	G	US-6,390,374	05-2002	Carper et al.	235/492
*	H	US-6,233,683	05-2001	Chan et al.	713/172
*	I	US-6,795,923	09-2004	Stern et al.	726/12
*	J	US-6,324,650	11-2001	Ogilvie, John W.L.	726/2
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	**Application/Control No.** 10381219	**Applicant(s)/Patent Under Reexamination** YACH ET AL.
‖‖‖‖‖‖‖‖‖ (barcode)	**Examiner** JEREMIAH AVERY	**Art Unit** 2431

SEARCHED

Class	Subclass	Date	Examiner
none	none	8/26/2010	JLA

SEARCH NOTES

Search Notes	Date	Examiner
Updated EAST Search	8/26/2010	JLA
Keyword Search within Class 711, subclass 100, Class 713, subclasses 1, 176, 187 and 189. Class 395, subclass 682 and Class 719, subclass 328	8/25/2010	JLA
Updated Inventor Search	8/25/2010	JLA

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
none	none	8/26/2010	JLA

	Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
		10381219	YACH ET AL.
		Examiner	Art Unit
		JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE									
Final	Original	08/26/2010									
	1	-									
	2	-									
	3	-									
	4	-									
	5	-									
	6	-									
	7	-									
	8	-									
	9	-									
	10	-									
	11	-									
	12	-									
	13	-									
	14	-									
	15	-									
	16	-									
	17	-									
	18	-									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
	26	-									
	27	-									
	28	-									
	29	-									
	30	-									
	31	-									
	32	-									
	33	-									
	34	-									
	35	-									
	36	-									

U.S. Patent and Trademark Office   Part of Paper No. : 20100824

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant	☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE								
Final	Original	08/26/2010								
	37	-								
	38	-								
	39	-								
	40	-								
	41	-								
	42	-								
	43	-								
	44	-								
	45	-								
	46	-								
	47	-								
	48	-								
	49	-								
	50	-								
	51	-								
	52	-								
	53	-								
	54	-								
	55	-								
	56	-								
	57	-								
	58	-								
	59	-								
	60	-								
	61	-								
	62	-								
	63	-								
	64	-								
	65	-								
	66	-								
	67	-								
	68	-								
	69	-								
	70	-								
	71	-								
	72	-								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	**Rejected**	-	**Cancelled**	N	**Non-Elected**	A	**Appeal**
=	**Allowed**	÷	**Restricted**	I	**Interference**	O	**Objected**

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	08/26/2010								
	73	-								
	74	-								
	75	-								
	76	-								
	77	-								
	78	-								
	79	-								
	80	-								
	81	-								
	82	-								
	83	-								
	84	-								
	85	-								
	86	-								
	87	-								
	88	-								
	89	-								
	90	-								
	91	-								
	92	-								
	93	-								
	94	-								
	95	-								
	96	-								
	97	-								
	98	-								
	99	-								
	100	-								
	101	-								
	102	-								
	103	-								
	104	-								
	105	-								
	106	-								
	107	-								
	108	-								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant	☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE									
Final	Original	08/26/2010									
	109	-									
	110	-									
	111	-									
	112	-									
	113	-									
	114	-									
	115	-									
	116	-									
	117	-									
	118	-									
	119	-									
	120	-									
	121	-									
	122	-									
	123	-									
	124	-									
	125	-									
	126	-									
	127	-									
	128	-									
	129	-									
	130	-									
	131	-									
	132	-									
	133	-									
	134	-									
	135	-									
	136	-									
	137	-									
	138	-									
	139	-									
	140	-									
	141	-									
	142	-									
	143	-									
	144	-									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE									
Final	Original	08/26/2010									
	145	-									
	146	-									
	147	-									
	148	-									
	149	-									
	150	-									
	151	-									
	152	-									
	153	-									
	154	-									
	155	-									
	156	-									
	157	-									
	158	-									
	159	-									
	160	-									
	161	-									
	162	-									
	163	-									
	164	-									
	165	-									
	166	✓									
	167	✓									
	168	✓									
	169	✓									
	170	✓									
	171	✓									
	172	✓									
	173	✓									
	174	✓									
	175	✓									
	176	✓									
	177	✓									
	178	✓									
	179	✓									
	180	✓									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	08/26/2010								
	181	✓								
	182	✓								
	183	✓								
	184	✓								
	185	✓								
	186	✓								
	187	✓								
	188	✓								
	189	✓								
	190	✓								
	191	✓								
	192	✓								
	193	✓								
	194	✓								
	195	✓								
	196	✓								
	197	✓								
	198	✓								
	199	✓								
	200	✓								
	201	✓								
	202	✓								
	203	✓								
	204	✓								
	205	✓								
	206	✓								
	207	✓								
	208	✓								
	209	✓								
	210	✓								
	211	✓								
	212	✓								
	213	✓								
	214	✓								
	215	✓								
	216	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE									
Final	Original	08/26/2010									
	217	✓									
	218	✓									
	219	✓									
	220	✓									
	221	✓									
	222	✓									
	223	✓									
	224	✓									
	225	✓									
	226	✓									
	227	✓									
	228	✓									
	229	✓									
	230	✓									
	231	✓									
	232	✓									
	233	✓									
	234	✓									
	235	✓									
	236	✓									
	237	✓									
	238	✓									
	239	✓									
	240	✓									
	241	✓									
	242	✓									
	243	✓									
	244	✓									
	245	✓									
	246	✓									
	247	✓									
	248	✓									
	249	✓									
	250	✓									
	251	✓									
	252	✓									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	**Rejected**	-	**Cancelled**	N	**Non-Elected**	A	**Appeal**
=	**Allowed**	÷	**Restricted**	I	**Interference**	O	**Objected**

☐ Claims renumbered in the same order as presented by applicant	☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE
Final	**Original**	08/26/2010								
	253	✓								
	254	✓								
	255	✓								
	256	✓								
	257	✓								
	258	✓								
	259	✓								
	260	✓								
	261	✓								
	262	✓								
	263	✓								
	264	✓								
	265	✓								

PTO/SB/08A (09-06)
Approved for use through 03/31/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Use as many sheets as necessary)

Sheet | 1 | of | 1

Complete if Known

Application Number	10/381,219
Filing Date	March 20, 2003
First Named Inventor	David P. Yach
Art Unit	2431
Examiner Name	Jeremiah L. Avery
Attorney Docket Number	555255-012423

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number — Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 20020112078	08-15-2002	Yach, David	
		US- 20020128036	09-12-2002	Yach, et al.	
		US- 20030026231	02-06-2003	Lazaridis, et al.	
		US- 20030159029	08-21-2003	Brown, et al.	
		US- 20040166834	08-26-2004	Omar, et al.	
		US- 20040170155	09-02-2004	Omar, et al.	
		US- 20040171369	09-02-2004	Little, et al.	
		US- 20040171374	09-02-2004	Little, et al.	
		US- 20040199665	10-07-2004	Omar, et al.	
		US- 20040202327	10-14-2004	Little, et al.	
		US- 20040205330	10-14-2004	Godfrey, et al.	
		US- 20050009502	01-13-2005	Little, et al.	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document — Country Code[3]-Number[4]-Kind Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]	
✓		AU9736815	02-19-1998	Intertrust Technologies Corp.		✓	Abstract only.

Examiner Signature	/Jeremiah Avery/	Date Considered	08/26/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

Substitute for form 1449/PTO	**Complete if Known**	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*	Application Number	10381219
	Filing Date	March 20, 2003
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
Sheet 2 of 2	Attorney Docket Number	555255-012423

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Rankl, Wolfgang, et al., Handbuch der Chipkarten, Aufbau - Funktionsweise - Einsatz von Smart Cards, Hanser, 1999 - in German.	

Examiner Signature	/Jeremiah Avery/	Date Considered	08/26/2010

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) near$ (software or application or program)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:16
L2	97	L1 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:16
L3	97	L1 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:18
L4	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and (@ad<"20000921"	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:29

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (1 of 13)8/26/2010 3:10:43 PM

Page 760 of 1415

		@pd<"20000921" @rlad<"20000921")				
L5	98	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$4 near interface)) and ((deny$ or denies or denial) same (digital near signature)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:30
L6	98	L5 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:30
L7	101328	(@ad<"20000921" @pd<"20000921" @rlad<"20000921") and ((eras $ or purg$ or delet$ or expung $ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$ or flush$) near$ (software or application or program or trojan)) and (authentic$ or verify$ or verification) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:36
L8	32	l7 and ((digital near signature) same ((unauthentic$ or unverify$ or unverifi$) or ("not" near (authentic$ or verify$ or verification))))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:40
L9	992	l7 and ((unauthentic$ or unverify$ or unverifi$) or ("not" near (authentic$ or verify$ or verification)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:47
L10	484	l9 and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:48

Page 761 of 1415

L11	481	l10 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:49
L12	126	l11 and (API or (application near program$4 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:50
S1	8	(digital near signature) and (authentic$ or verify$ or verificat$) and (virtual near machine) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:23
S2	8	S1 and (portab$ or mobile or handheld or laptop or pda or cell or cellular)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S3	4	S2 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S4	35	(digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:28
S5	737	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33

file:///Cl/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (3 of 13)8/26/2010 3:10:43 PM

Page 762 of 1415

S6	41	S5 and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) same (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33
S7	30	S6 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:37
S8	30	S7 and access$	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S9	30	S8 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S10	2	S9 and ((secure near hash near algorithm) or SHA?1)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:39
S11	1	S10 and public and private	US-PGPUB; USPAT	OR	ON	2009/02/20 11:40
S12	31	S6 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:46
S13	31	S12 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:47
S14	31	S13 and ((public or private) same key)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:48
S15	0	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) same (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (4 of 13)8/26/2010 3:10:43 PM

Page 763 of 1415

S16	30	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) and (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53
S17	28	S16 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:07
S18	118	S5 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S19	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S20	56	S19 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:26
S21	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S22	56	S21 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S23	40	S22 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:28
S24	55	S22 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:32
S25	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (5 of 13)8/26/2010 3:10:43 PM

Page 764 of 1415

S26	119	S25 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S27	62	S26 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S28	16	S27 and (SIM or (subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S29	30	S25 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda) and (SIM or (Subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:14
S30	16	S29 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:24
S31	9	S28 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:28
S32	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S33	119	S32 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S34	62	S33 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (6 of 13)8/26/2010 3:10:43 PM

Page 765 of 1415

S35	57	S34 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S36	56	S35 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S37	56	S36 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S38	36	S37 and (digital near signature) and ((authentic$ or verify$ or verification) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S39	0	S38 and (signature near (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S40	3	S38 and (signature same (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:53
S41	40	S37 and (digital near signature) and ((authentic$ or verify$ or verification or valid $) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57
S42	6	S41 and (signature same (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57
S43	757	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S44	130	S43 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S45	72	S44 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S46	65	S45 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29

S47	64	S46 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S48	64	S47 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S49	48	S48 and (digital near signature) and ((authentic$ or verify$ or verification or valid $) near signature)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S50	6	S49 and (signature same (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
S51	11	S49 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat $) near (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:37
S52	21	S49 and (virtual near machine)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:38
S53	25263	(@ad<"20000921" @prad<"20000921") and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) near (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:13
S54	31	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial or unauthentic$ or unverif$4) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:15
S55	0	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial or unauthentic$ or unverif$4) near (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:53

S56	62	S53 and (digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:53
S57	31	S56 and ((deny$ or denies or denial or unauthentic$ or unverif$4) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5))	US-PGPUB; USPAT	OR	ON	2009/11/24 10:55
S58	94	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) same (software or application or program)) and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:48
S59	14	S58 and (virtual near machine)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:50
S60	61	S58 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:51
S61	61	S60 and ($2crypt$ or $2cipher$)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:51
S62	61	S61 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:52

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (9 of 13)8/26/2010 3:10:43 PM

Page 768 of 1415

S63	1	S62 and S59	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 14:52
S64	13286	711/100.ccls. or 713/1.ccls. or 713/176.ccls. or 713/187.ccls. or 713/189.ccls. or 395/682. cxr. and (API or (application near programming near interface)) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:04
S65	4049	S64 and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:04
S66	2341	S65 and ((digital near signature) near$4 (authentic$ or verify$ or verificat$))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:06
S67	343	(719/328.ccls. or 711/100. ccls. or 713/1.ccls. or 713/176. ccls. or 713/187.ccls. or 713/189.ccls. or 395/682.cxr.) and (API or (application near programming near interface)) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S68	343	S67 and ((digital near signature) near$4 (authentic$ or verify$ or verificat$))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S69	115	S68 and (@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S70	115	S69 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:07
S71	85	S70 and ((public and private) near key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:08

S72	94	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$5 near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5) same (software or application or program)) and ((@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:11
S73	76	S71 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:12
S74	97	(digital near signature) and (authentic$ or verify$ or verificat$) and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg $ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) near$ (software or application or program)) and ((@ad<"20000921" @pd<"20000921" @rlad<"20000921")	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:28
S75	1	S74 and ((hash$ or (one?way or (one near way)) or abridg$) near key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:30
S76	0	S70 and (((deny or denying or denial or restrict$ or prohibit$) near access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:37

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (11 of 13)8/26/2010 3:10:43 PM

Page 770 of 1415

S77	53	S70 and (((deny or denying or denial or restrict$ or prohibit$) near$ access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:37
S78	53	S70 and (((deny or denying or denied or denial or restrict$ or prohibit$) near$ access$) same (API or (Application near program$5 near interface)))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:38
S79	53	S78 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:40
S80	43	S79 and ((hash$ or (one?way or (one near way)) or abridg$) same key)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:40
S81	53	S78 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:54
S82	97	S74 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or dump$) near$ (software or application or program))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 15:55
S83	773	(YACH-DAVID-P.in. or BROWN-MICHAEL-S.in. or LITTLE-HERBERT-A.in.)	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:21
S84	180	(digital near signature) and (authentic$ or verify$ or verificat$) and S83	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:22
S85	177	S84 and access$	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:22
S86	41	S84 and (mobile or portable or cell or cellular or phone or telephone or laptop or PDA or (pocket near pc) or (personal near digital near assistant)) and ((API) or (Application near program$5 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2010/08/24 16:24

file:///C/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (12 of 13)8/26/2010 3:10:43 PM

Page 771 of 1415

8/ 26/ 2010 3:10:32 PM
C:\ Documents and Settings\ javery\ My Documents\ EAST\ Workspaces\ 10381219.wsp

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (13 of 13)8/26/2010 3:10:43 PM

Page 772 of 1415

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	:	Yach, et al.
Title	:	SOFTWARE CODE SIGNING SYSTEM AND METHOD
Application No.	:	10/381,219
Filing Date	:	March 20, 2003
Confirmation No.	:	9761
Examiner	:	Jeremiah L. Avery
Group Art Unit	:	2431
Attorney Docket	:	555255-012423

Commissioner for Patents
P.O. Box 1450
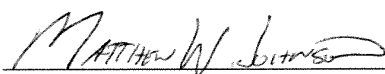Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

In compliance with 37 CFR 1.56, lists of patent references and documents are set forth on the attached PTO-1449 forms.

Pursuant to 37 CFR 1.98(a)(3), the Rankl reference is in the German language and it is understood to be relevant based on its citation in an opposition in a related European Patent Office application: EP20010973901.

It is believed that no fee is due, however in the event that a fee is due, please charge Jones Day's Deposit Account No. 501432, Ref. 555255-012423.

Respectfully submitted,

Matthew W. Johnson (Reg. No. 59,108)
JONES DAY
One Mellon Center, 45th Floor
500 Grant Street
Pittsburgh, PA 15219
(412) 391-3939

Substitute for form 1449/PTO	**Complete if Known**	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*	Application Number	10/381,219
	Filing Date	March 20, 2003
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
Sheet 1 of 1	Attorney Docket Number	555255-012423

U. S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number — Number-Kind Code[2] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US- 20020112078	08-15-2002	Yach, David	
		US- 20020128036	09-12-2002	Yach, et al.	
		US- 20030026231	02-06-2003	Lazaridis, et al.	
		US- 20030159029	08-21-2003	Brown, et al.	
		US- 20040166834	08-26-2004	Omar, et al.	
		US- 20040170155	09-02-2004	Omar, et al.	
		US- 20040171369	09-02-2004	Little, et al.	
		US- 20040171374	09-02-2004	Little, et al.	
		US- 20040199665	10-07-2004	Omar, et al.	
		US- 20040202327	10-14-2004	Little, et al.	
		US- 20040205330	10-14-2004	Godfrey, et al.	
		US- 20050009502	01-13-2005	Little, et al.	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document — Country Code[3]-Number[4]-Kind Code[5] *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]	
✓		AU9736815	02-19-1998	Intertrust Technologies Corp.		✓	Abstract only.

Examiner Signature		Date Considered	

Page 774 of 1415

Substitute for form 1449/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*				Application Number	10381219
				Filing Date	March 20, 2003
				First Named Inventor	David P. Yach
				Art Unit	2431
				Examiner Name	Jeremiah L. Avery
Sheet	2	of	2	Attorney Docket Number	555255-012423

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Rankl, Wolfgang, et al., Handbuch der Chipkarten, Aufbau - Funktionsweise - Einsatz von Smart Cards, Hanser, 1999 - in German.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.
This collection of information is required by 37 CFR 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

(12) PATENT ABSTRACT   **(11)** Document No. **AU-A-36815/97**

(19) AUSTRALIAN PATENT OFFICE

(54)   Title
SYSTEMS AND METHODS USING CRYPTOGRAPHY TO PROTECT SECURE COMPUTING
ENVIRONMENTS

International Patent Classification(s)
(51)⁶   G06F 012/14   H04L 009/30   H04L 009/32

(21)   Application No. : 36815/97   (22) Application Date : 04/09/97

(30)   Priority Date

(31)   Number   (32)   Date   (33)   Country
08/689754   12/08/96   US UNITED STATES OF AMERICA

(43)   Publication Date : 19/02/98

(71)   Applicant(s)
INTERTRUST TECHNOLOGIES CORP.

(72)   Inventor(s)
VICTOR H SHEAR; W. OLIN SIBERT; DAVID M VAN WIE

(74)   Attorney or Agent
DAVIES COLLISON CAVE , 1 Little Collins Street, MELBOURNE VIC 3000

(57)

Secure computation environments are protected from
bogus or rogue load modules, executables and other data
elements through use of digital signatures, seals and
certificates issued by a verifying authority. A verifying
authority - which may be a trusted independent third party —
tests the load modules or other executables to verify that their
corresponding specifications are accurate and complete, and
then digitally signs the load module or other executable based
on tamper resistance work factor classification. Secure
computation environments with different tamper resistance
work factors use different verification digital signature
authentication techniques (e.g., different signature algorithms

.../2

and/or signature verification keys) — allowing one tamper
resistance work factor environment to protect itself against
load modules from another, different tamper resistance work
factor environment. Several dissimilar digital signature
algorithms may be used to reduce vulnerability from algorithm
compromise, and subsets of multiple digital signatures may be
used to reduce the scope of any specific compromise.

36 815 /97

FIG. 2 Verifying Load Modules
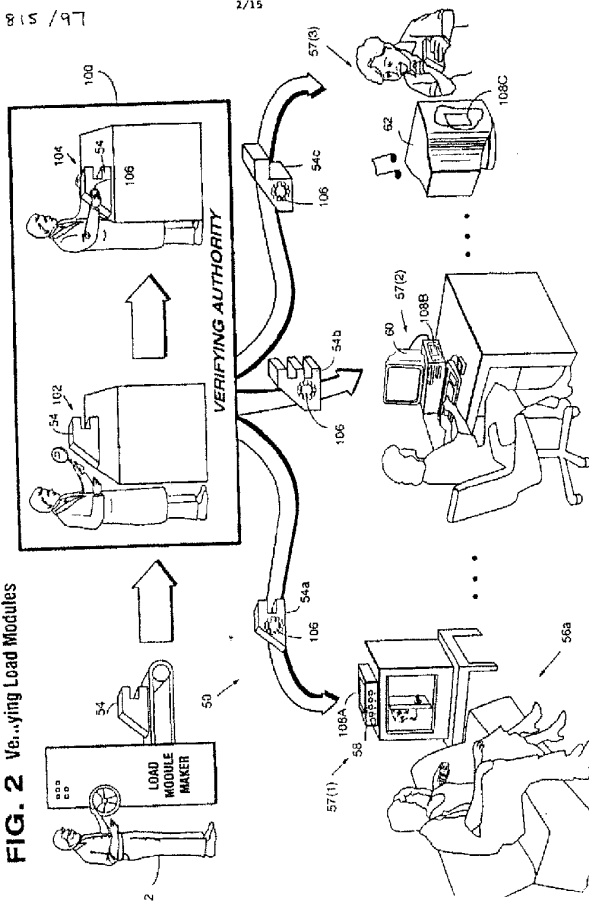
Electronic Acknowledgement Receipt

EFS ID:	8094443
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Stephen D. Scanlon/Matthew W. Johnson
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	555255012423
Receipt Date:	27-JUL-2010
Filing Date:	20-MAR-2003
Time Stamp:	10:19:50
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	012423_ids.pdf	45701 a6cdfaa7bd1c66987068d75d5f2795c784c154d1	no	1

Warnings:

Information:

This is not an USPTO supplied IDS fillable form					
2	Information Disclosure Statement (IDS) Filed (SB/08)	IDS_012423.pdf	120898 b0076e21163d93750bb6cd6fe72b283112c f2cdb	no	2

Warnings:

Information:

This is not an USPTO supplied IDS fillable form					
3	Foreign Reference	DOC003.pdf	58416 12d962e94609e9b9857c904103345c71bab a2304	no	3

Warnings:

Information:

4	NPL Documents	Rankl.pdf	2188336 8c30e1a3767550edbc31db701c0fe6e774f0 0ea5	no	29

Warnings:

Information:

Total Files Size (in bytes):	2413351

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Doc code: RCEX
Doc description: Request for Continued Examination (RCE)
PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)

Application Number	10/381,219	Filing Date	2003-03-20	Docket Number (if applicable)	555255-012423	Art Unit	2431
First Named Inventor	Yach et al.			Examiner Name	J. Avery		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☒ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

  ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____

  ☒ Other ___Amendment previously filed on March 22, 2010_____

☒ Enclosed

  ☒ Amendment/Reply

  ☐ Information Disclosure Statement (IDS)

  ☐ Affidavit(s)/ Declaration(s)

  ☒ Other ___Petition for Extension of Time_____

MISCELLANEOUS

☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

☐ Other

FEES

The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No   501432

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature
☐ Applicant Signature

Signature of Registered U.S. Patent Practitioner			
Signature	/Arriènne M. Lezak/	Date (YYYY-MM-DD)	2010-03-30
Name	Arriènne M. Lezak	Registration Number	51943

EFS - Web 2.1.15

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2009** *(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number (Optional) 555255-012423
Application Number 10/381,219	Filed March 20, 2003

For   Software Code Signing System And Method

Art Unit 2431	Examiner J. Avery

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Small Entity Fee	
☐	One month (37 CFR 1.17(a)(1))	$130	$65	$_____
☐	Two months (37 CFR 1.17(a)(2))	$490	$245	$_____
☑	Three months (37 CFR 1.17(a)(3))	$1110	$555	$620.00
☐	Four months (37 CFR 1.17(a)(4))	$1730	$865	$_____
☐	Five months (37 CFR 1.17(a)(5))	$2350	$1175	$_____

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 501432   .

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the   ☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
   Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☑ attorney or agent of record. Registration Number 51,943

☐ attorney or agent under 37 CFR 1.34.
   Registration number if acting under 37 CFR 1.34 _____

/Arriènne M. Lezak/	March 30, 2010
Signature	Date
Arriènne M. Lezak	(650) 687-4163
Typed or printed name	Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ Total of _____ forms are submitted.

Electronic Patent Application Fee Transmittal

Application Number:	10381219
Filing Date:	20-Mar-2003
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Filer:	Arrienne Monique Lezak
Attorney Docket Number:	555255012423
Filed as Large Entity	

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 3 months with $490 paid	1253	1	620	620

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
			Total in USD ($)	**1430**

Electronic Acknowledgement Receipt

EFS ID:	7318799
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Arrienne Monique Lezak
Filer Authorized By:	
Attorney Docket Number:	555255012423
Receipt Date:	30-MAR-2010
Filing Date:	20-MAR-2003
Time Stamp:	17:02:52
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1430
RAM confirmation Number	3750
Deposit Account	501432
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	
Charge any Additional Fees required under 37 C.F.R. 1.492 (National application filing, search, and examination fees)	
Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		DOC211.pdf	125037 91eb7ec396ac0bcc56e7ef91f3e3a0f9b4296b61	yes	3

Multipart Description/PDF files in .zip description		
Document Description	Start	End
Request for Continued Examination (RCE)	1	2
Extension of Time	3	3

Warnings:

Information:

2	Fee Worksheet (PTO-875)	fee-info.pdf	31898 da919c474ee2e310ceeb1b8f5086f071106c18fc	no	2

Warnings:

Information:

	Total Files Size (in bytes):	156935

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 10/381,219	Filing Date 03/20/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

OTHER THAN

	(Column 1)	(Column 2)	SMALL ENTITY ☐	OR	SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =		X $ =	
☐APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))						
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	

APPLICATION AS AMENDED – PART II

OTHER THAN

SMALL ENTITY   OR   SMALL ENTITY

AMENDMENT

03/30/2010	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 109	Minus	** 109	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 12	Minus	***12	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))							OR		
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))							OR		
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/CATHERINE d. SMITH/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of :   David P. Yach; Michael S. Brown; Herbert A. Little

Internat'l. Appl'n. No. :   PCT/CA01/01344

Internat'l. Filing Date :   09/20/2001

U.S. Serial No.   :   10/381,219

U.S. Filing Date   :   03/20/2003

Title   :   Software Code Signing System And Method

Art Unit   :   2431

Examiner   :   J. Avery

Docket No.   :   555255-012423

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND INTERVIEW SUMMARY IN REPLY TO THE ACTION
MAILED OCTOBER 29, 2009

Please amend the above-identified application as follows:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	David P. Yach; Michael S. Brown; Herbert A. Little
Internat'l. Appl'n. No.	:	PCT/CA01/01344
Internat'l. Filing Date	:	09/20/2001
U.S. Serial No.	:	10/381,219
U.S. Filing Date	:	03/20/2003
Title	:	Software Code Signing System And Method
Art Unit	:	2431
Examiner	:	J. Avery
Docket No.	:	555255-012423

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND INTERVIEW SUMMARY IN REPLY TO THE ACTION

MAILED OCTOBER 29, 2009

Please amend the above-identified application as follows:

Page 790 of 1415

Amendments to the claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-165. (Cancelled).

166.   (New) A mobile device containing software instructions which when executed on the mobile device cause the mobile device to perform operations for controlling access to an application platform, the operations comprising:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature;

receiving a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

167.   (New) The mobile device of claim 166, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

168. (New) The mobile device of claim 166, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

169. (New) The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

170. (New) The mobile device of claim 166, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

171. (New) The mobile device of claim 166, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

172. (New) The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

173. (New) The mobile device of claim 166, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

174. (New) The mobile device of claim 166, wherein the application execution manager is implemented by a virtual machine (VM).

175. (New) The mobile device of claim 166, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

176. (New) The mobile device of claim 166, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

177. (New) The mobile device of claim 166, wherein the digital signature is generated by a code signing authority and included with the software application.

178. (New) The mobile device of claim 166, wherein the operations for controlling access to the application platform further comprise:

displaying a description string when the software application attempts to access the sensitive API.

179. (New) The mobile device of claim 166, wherein the application platform comprises an operating system.

180. (New) The mobile device of claim 166, wherein the application platform is on the mobile device, and wherein the application platform includes mobile device hardware.

181. (New) The mobile device of claim 166, wherein the application platform comprises a cryptographic module.

182. (New) The mobile device of claim 166, wherein the application platform comprises a data store.

183. (New) The mobile device of claim 166, wherein the application platform comprises a proprietary data model.

184. (New) The mobile device of claim 166, wherein the application platform comprises an input and output controller.

185. (New) The mobile device of claim 166, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

186. (New) The mobile device of claim 185, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

187. (New) The mobile device of claim 186, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

188. (New) The mobile device of claim 166, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

189. (New) The mobile device of claim 166, wherein the digital signature and the signature identification correspond to a mobile device type.

190. (New) The mobile device of claim 166, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

191. (New) A system for controlling access to an application platform, comprising:
   one or more processors;

one or more computer-readable storage mediums containing software instructions executable on the one or more processors to cause the one or more processors to perform operations including:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature;

receiving a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

192.   (New) The system of claim 191, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

193.   (New) The system of claim 191, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

194. (New) The system of claim 191, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

195. (New) The system of claim 191, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

196. (New) The system of claim 191, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

197. (New) The system of claim 191, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

198. (New) The system of claim 191, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

199. (New) The system of claim 191, wherein the application execution manager is implemented by a virtual machine (VM).

200. (New) The system of claim 191, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

201.   (New) The system of claim 191, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

202.   (New) The system of claim 191, further comprising:

a code signing authority, wherein the code signing authority determines whether the software application should be given access to a sensitive API, and based upon a determination that the software application should be given access to a sensitive API, the code signing authority accepts the software application and generates a digital signature that is included with the software application.

203.   (New) The system of claim 191, wherein the operations performed by the one or more processors further comprise:

displaying a description string when the software application attempts to access the sensitive API.

204.   (New) The system of claim 191, wherein the application platform comprises an operating system.

205.   (New) The system of claim 191, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

206.   (New) The system of claim 191, wherein the application platform comprises a cryptographic module.

207.   (New) The system of claim 191, wherein the application platform comprises a data store.

208. (New) The system of claim 191, wherein the application platform comprises a proprietary data model.

209. (New) The system of claim 191, wherein the application platform comprises an input and output controller.

210. (New) The system of claim 191, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

211. (New) The system of claim 210, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

212. (New) The system of claim 211, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

213. (New) The system of claim 191, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

214. (New) The system of claim 191, wherein the digital signature and the signature identification correspond to a mobile device type.

215. (New) The system of claim 191, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

216. (New) A computer-readable storage medium encoded with instructions that when executed on one or more processors within a computer system perform a method for controlling access to an application platform, the method comprising:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature;

receiving a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

217. (New) The computer-readable storage medium of claim 216, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

218. (New) The computer-readable storage medium of claim 216, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

219. (New) The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

220.   (New) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

221.   (New) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

222.   (New) The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

223.   (New) The computer-readable storage medium of claim 216, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

224.   (New) The computer-readable storage medium of claim 216, wherein the application execution manager is implemented by a virtual machine (VM).

225.   (New) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying a private key to a first hash of the software application; and
        the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

226.   (New) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and
        the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital

signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

227. (New) The computer-readable storage medium of claim 216, wherein the digital signature is generated by a code signing authority and included with the software application.

228. (New) The computer-readable storage medium of claim 216, further comprising:
    displaying a description string when the software application attempts to access the sensitive API.

229. (New) The computer-readable storage medium of claim 216, wherein the application platform comprises an operating system.

230. (New) The computer-readable storage medium of claim 216, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

231. (New) The computer-readable storage medium of claim 216, wherein the application platform comprises a cryptographic module.

232. (New) The computer-readable storage medium of claim 216, wherein the application platform comprises a data store.

233. (New) The computer-readable storage medium of claim 216, wherein the application platform comprises a proprietary data model.

234. (New) The computer-readable storage medium of claim 216, wherein the application platform comprises an input and output controller.

235. (New) The computer-readable storage medium of claim 216, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

236. (New) The computer-readable storage medium of claim 235, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

237. (New) The computer-readable storage medium of claim 236, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

238. (New) The computer-readable storage medium of claim 216, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

239. (New) The computer-readable storage medium of claim 216, wherein the digital signature and the signature identification correspond to a mobile device type.

240. (New) The computer-readable storage medium of claim 216, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

241. (New) A method for controlling access to an application platform, comprising:

storing a plurality of application programming interfaces (APIs), wherein each API can be used to allow certain software applications to interact with the application platform, and wherein at least one API comprises a sensitive API accessible upon verification of a digital signature;

receiving, using one or more processors, a software application requesting access to a sensitive API, wherein the sensitive API is associated with a signature identifier and a public key;

using an application execution manager to:

determine whether the software application is signed, wherein a signed software application includes a digital signature and a corresponding signature identification,

based upon a determination that the software application is signed, determine whether the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, and

based upon a determination that the signature identifier of the sensitive API corresponds to the signature identification of the signed software application, use the public key associated with the sensitive API to verify authenticity of the digital signature of the signed software application; and

upon verifying the authenticity of the digital signature, using the sensitive API to allow the signed software application to interact with the application platform.

242.   (New) The method of claim 241, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is not executed.

243.   (New) The method of claim 241, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the software application is denied access to the sensitive API.

244.   (New) The method of claim 241, wherein the application platform is on a mobile device, and wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature identification, the application execution manager purges the software application from the mobile device.

245.   (New) The method of claim 241, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is not executed.

246.	(New) The mobile device of claim 241, wherein based upon a determination that the digital signature is not authenticated, the software application requesting access to the sensitive API is denied access to the sensitive API.

247.	(New) The method of claim 241, wherein the application platform is on a mobile device, and wherein based upon a determination that the digital signature is not authenticated, the application execution manager purges the software application requesting access to the sensitive API from the mobile device.

248.	(New) The method of claim 241, wherein a global signature is associated with each of the plurality of APIs; and wherein the global signature is verified prior to allowing the signed software application to interact with the application platform.

249.	(New) The method of claim 241, wherein the application execution manager is implemented by a virtual machine (VM).

250.	(New) The method of claim 241, wherein the digital signature is generated by applying a private key to a first hash of the software application; and

the digital signature is authenticated by generating a second hash of the software application to obtain a generated hash, applying the public key to the digital signature to obtain a recovered hash, and verifying that the generated hash and the recovered hash are the same.

251.	(New) The method of claim 241, wherein the digital signature is generated by applying a private key to a first abridged version of the software application; and

the digital signature is authenticated by generating a second abridged version of the software application to obtain a generated abridged version, applying the public key to the digital signature to obtain a recovered abridged version, and verifying that the generated abridged version and the recovered abridged version are the same.

252.	(New) The method of claim 241, further comprising:

determining by a code signing authority, whether the software application should be given access to a sensitive API, wherein based upon a determination that the software application should be given access to a sensitive API, the code signing authority accepts the software application and generates a digital signature that is included with the software application.

253. (New) The method of claim 241, wherein the operations for controlling access to the application platform further comprise:

displaying a description string when the software application attempts to access the sensitive API.

254. (New) The method of claim 241, wherein the application platform comprises an operating system.

255. (New) The method of claim 241, wherein the application platform is on a mobile device, and wherein the application platform includes mobile device hardware.

256. (New) The method of claim 241, wherein the application platform comprises a cryptographic module.

257. (New) The method of claim 241, wherein the application platform comprises a data store.

258. (New) The method of claim 241, wherein the application platform comprises a proprietary data model.

259. (New) The method of claim 241, wherein the application platform comprises an input and output controller.

260. (New) The method of claim 241, wherein the digital signature provides an audit trail identifying a developer of the software application requesting access to the sensitive API.

261.   (New) The method of claim 260, wherein a problematic software application is identified using the audit trail, and wherein the digital signature associated with the problematic software application is revocable.

262.   (New) The method of claim 261, wherein the digital signature associated with the problematic software application is revoked, and wherein the revoked digital signature is added to a signature revocation list.

263.   (New) The method of claim 241, wherein the authenticity of the digital signature is first verified each time the software application requesting access to the sensitive API is allowed to interact with the application platform.

264.   (New) The method of claim 241, wherein the digital signature and the signature identification correspond to a mobile device type.

265.   (New) The method of claim 241, wherein associating the sensitive API with the public key includes obtaining the public key from a public key repository.

REMARKS

Claims 57-165 are pending in this application. Claims 57, 62, 83, 92, 99, 103, 112, 124, 133, 143, 152, and 160 are in independent form. No claims are being amended. Claims 57-165 are being cancelled. Claims 166-265 are being newly added. No new matter has been added. Support for the newly added claims can be found within the applicant's specification at least at page 13, line 3 through page 15, line 13 and FIG. 3. Independent claims 166, 191, 216, and 241 are mobile device, system, and method claims reciting similar limitations.

Reconsideration and reexamination of the application is respectfully requested in light of the foregoing new claims and the following remarks.

Interview Summary

The applicant thanks Examiners Avery and Zia for the courtesy of the in-person interview on March 8, 2010. The time spent with the applicant's representatives Lisa Koh and Arriènne M. Lezak was greatly appreciated. During the interview, the specification was discussed in view of the cited references. Recommendations were made by the examiners. These recommendations have been incorporated into the new claims as noted above.

Claim Objections

Claim 112 was objected to because of certain informalities. Claim 112 has been cancelled.

Withdrawal of the claim objection is therefore respectfully requested.

Section 102 Rejections

Claims 57-71, 74-77, 79-87, 90-97, 99-108, 111-118, 121-127, 130-137, 140-147, 150-156, 159-162, and 165 were rejected under 35 U.S.C. §102(e) as allegedly anticipated by U.S. Patent No. 6,795,919 ("Gibbs").

To expedite prosecution, and in accordance with the examiners' recommendations, claims 57-71, 74-77, 79-87, 90-97, 99-108, 111-118, 121-127, 130-137, 140-147, 150-156, 159-162, and 165 have been cancelled.

Withdrawal of the rejection under 35 U.S.C. §102(e) is therefore respectfully requested.

Page 807 of 1415

Section 103 Rejections

Claim 78 was rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Gibbs in view of U.S. Patent No. 6,584,376 ("Van Kommer").

Claims 72, 73, 88, 89, 98, 109, 110, 119, 120, 128, 129, 138, 139, 148, 149, 157, 158, 163, and 164 were rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Gibbs in view of U.S. Patent No. 6,587,837 ("Spagna").

To expedite prosecution, and in accordance with the examiners' recommendations, claims 72, 73, 78, 88, 89, 98, 109, 110, 119, 120, 128, 129, 138, 139, 148, 149, 157, 158, 163, and 164 have been cancelled.

Withdrawal of the rejection under 35 U.S.C. §103(a) is therefore respectfully requested.

New Claims 166-265

Support for new claims 166-265 is noted above. The applicant respectfully submits that Gibbs, Van Kommer, and Spagna, alone or in combination, do not teach or describe each and every aspect of new claims 166-265. Thus, the applicant submits that new claims 166-265 are allowable.

Conclusion

The applicant respectfully requests that all pending claims be allowed.

By responding in the foregoing remarks only to particular positions taken by the examiner, the applicant does not acquiesce with other positions that have not been explicitly addressed. In addition, the applicant selecting some particular arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist. Finally, the applicant's decision to amend or cancel any claim should not be understood as implying that the applicant agrees with any positions taken by the examiner with respect to that claim or other claims.

Please apply any charges or credits to Deposit Account No. 50-1432, Reference No. 555255-012423.

Respectfully submitted,

Date: __March 22, 2010__   By: __/Arriènne M. Lezak/_____
                                        Arriènne M. Lezak
                                        Reg. No. 51,943
                                        JONES DAY
                                        1755 Embarcadero Road
                                        Palo Alto, CA 94303
                                        (650) 687-4163

Electronic Patent Application Fee Transmittal

Application Number:	10381219
Filing Date:	20-Mar-2003
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Filer:	Stephen D. Scanlon/Arrienne M. Lezak
Attorney Docket Number:	555255012423
Filed as Large Entity	

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 2 months with $0 paid	1252	1	490	490

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**490**

Electronic Acknowledgement Receipt

EFS ID:	7258230
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Stephen D. Scanlon/Arrienne M. Lezak
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	555255012423
Receipt Date:	22-MAR-2010
Filing Date:	20-MAR-2003
Time Stamp:	16:47:18
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$490
RAM confirmation Number	3565
Deposit Account	501432
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	
Charge any Additional Fees required under 37 C.F.R. 1.492 (National application filing, search, and examination fees)	
Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Extension of Time	DOC019.pdf	56113 7da7c5db4d1aa2d0c59c245657b727801ac dbcce	no	1

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
2	Amendment After Final	DOC020.pdf	744296 a8a612b27dcc80ea83bd961e02c458a5be7 5b9d4	no	20

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
3	Fee Worksheet (PTO-875)	fee-info.pdf	30122 e597d038fd0c3b59eb45e76b1fcf98e32df0 2c31	no	2

Warnings:

Information:

	Total Files Size (in bytes):	830531

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) **FY 2009** *(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)*	Docket Number (Optional) 555255-012423
Application Number 10/381,219	Filed March 20, 2003

For   Software Code Signing System And Method

Art Unit 2431	Examiner J. Avery

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

		Fee	Small Entity Fee	
☐	One month (37 CFR 1.17(a)(1))	$130	$65	$_____
☑	Two months (37 CFR 1.17(a)(2))	$490	$245	$ 490.00
☐	Three months (37 CFR 1.17(a)(3))	$1110	$555	$_____
☐	Four months (37 CFR 1.17(a)(4))	$1730	$865	$_____
☐	Five months (37 CFR 1.17(a)(5))	$2350	$1175	$_____

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 50-1432 .

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest. See 37 CFR 3.71.
      Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).

☑ attorney or agent of record. Registration Number 51,943

☐ attorney or agent under 37 CFR 1.34.
      Registration number if acting under 37 CFR 1.34 _____

/Arrienne M. Lezak/	March 22, 2010
Signature	Date
Arrienne M. Lezak	(650) 687-4163
Typed or printed name	Telephone Number

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☑ Total of ___1___ forms are submitted.

Page 814 of 1415

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
10/381,219	03/20/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

			SMALL ENTITY ☐	OR	OTHER THAN SMALL ENTITY	
	(Column 1)	(Column 2)				
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))						
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	

APPLICATION AS AMENDED – PART II

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
AMENDMENT **03/22/2010**	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 100	Minus	** 109	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 4	Minus	***12	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

	(Column 1)		(Column 2)	(Column 3)					
AMENDMENT	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/CATHERINE d. SMITH/

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

89441   7590   03/12/2010
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
03/12/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dlpejeau@jonesday.com
portfolioprosecution@rim.com

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Interview Summary	10/381,219	YACH ET AL.
	Examiner	**Art Unit**	
	JEREMIAH AVERY	2431	

All participants (applicant, applicant's representative, PTO personnel):

(1) _Jeremiah Avery_.   (3)_Arrienne Lezak, Reg. No. 51,943_.

(2) _Syed Zia_.   (4)_Lisa Koh, Reg. No. 43,725_.

  Date of Interview: _08 March 2010_.

  Type: a)☐ Telephonic b)☐ Video Conference
          c)☒ Personal [copy given to: 1)☐ applicant   2)☒ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes   e)☒ No.
    If Yes, brief description: _____.

Claim(s) discussed: _N/A_.

Identification of prior art discussed: _United States Patent No. 6,795,919 to Gibbs et al., hereinafter Gibbs_.

Agreement with respect to the claims f)☐ was reached. g)☒ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: _The specification was discussed in view of the claimed invention. The inventors discussed how their claimed invention distinguishes itself over Gibbs. No agreement with regards to the allowability of claims at this time_
.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

/Syed Zia/	/Jeremiah Avery/
Primary Examiner, Art Unit 2431	Examiner, Art Unit 2431

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

_____

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

89441   7590   12/03/2009

Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
12/03/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dlpejeau@jonesday.com
portfolioprosecution@rim.com

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Interview Summary	10/381,219	YACH ET AL.
	Examiner	Art Unit	
	JEREMIAH AVERY	2431	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Jeremiah Avery*.   (3)_____.

(2) *Matthew W. Johnson, Reg. No. 59,108*.   (4)_____.

   Date of Interview: *24 November 2009*.

   Type: a)☒ Telephonic   b)☐ Video Conference
          c)☐ Personal [copy given to: 1)☐ applicant   2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes   e)☒ No.
   If Yes, brief description: _____.

Claim(s) discussed: *57 and 59*.

Identification of prior art discussed: *United States Patent No. 6,795,919 to Gibbs et al., hereinafter Gibbs*.

Agreement with respect to the claims f)☐ was reached.   g)☒ was not reached.   h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Discussed the rejection of claims 57 and 59 pertaining to the purging of "the software application". The Examiner will consider the Applicant's remarks upon receiving a formal written response*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

/Jeremiah Avery/	/William R. Korzuch/
Examiner, Art Unit 2431	Supervisory Patent Examiner, Art Unit 2431

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

_____

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
  – Application Number (Series Code and Serial Number)
  – Name of applicant
  – Name of examiner
  – Date of interview
  – Type of interview (telephonic, video-conference, or personal)
  – Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
  – An indication whether or not an exhibit was shown or a demonstration conducted
  – An identification of the specific prior art discussed
  – An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
  – The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
  1) A brief description of the nature of any exhibit shown or any demonstration conducted,
  2) an identification of the claims discussed,
  3) an identification of the specific prior art discussed,
  4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
  5) a brief identification of the general thrust of the principal arguments presented to the examiner,
      (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
  6) a general indication of any other pertinent matters discussed, and
  7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

89441   7590   10/29/2009
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
10/29/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

dlpejeau@jonesday.com
portfolioprosecution@rim.com

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _15 June 2009_.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _57-165_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _57-165_ is/are rejected.

7)☒ Claim(s) _112_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _20 March 2003_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☒ All   b)☐ Some *   c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

I. Claims 57-165 have been examined.

II. Responses to Applicant's remarks have been given.

Response to Arguments

1.   Applicant's arguments, see page 30, filed 06/15/09, with respect to the objections
to claims 86, 92, 105 and 132 have been fully considered and are persuasive. The
objection to said claims has been withdrawn. However, the objection to claim 112 is
hereby maintained due to the term "identificaters" remaining in the preamble.

2.   Applicant's arguments, see page 30, filed 06/15/09, with respect to the 35 U.S.C.
112, second paragraph rejection of claims 82, 133, 143, 152 and 160 have been fully
considered and are persuasive. The 35 U.S.C. 112, second paragraph rejection to said
claims has been withdrawn.

3.   Applicant's arguments, see page 31, filed 06/15/09, with respect to the 35 U.S.C.
101 rejection of claims 112-132 have been fully considered and are persuasive. The 35
U.S.C. 101 rejection to said claims has been withdrawn.

4.   With regards to the Applicant's argument that "there is no disclosure that an API
has an associated signature identifier as required in claim 57", the Examiner maintains
the below-cited grounds of rejection and further asserts that Gibbs discloses this claim
limitation within, but not limited to, column 7, lines 35-38 and 51-65. The "authenticated
message server 428" within Gibbs initiates a request for a "unique digital signature 132"
and the request "identifies a particular service id 104 for which a unique digital signature
132 is desired". The Examiner broadly interprets the "associated signature identifier" to

pertain to the "service id 104" within Gibbs since the request comes from the

"authenticated message server" which contains an "application programming interface

('API')" (as cited within column 5, lines 40-45). There is a correlation between the

"unique signature 132" and the "service id 104", hence disclosing the claimed "an

application programming interface (API) having an associated signature identifier".

5.   With regards to the claim language of claim 57 pertaining to "a virtual machine

that verifies the authenticity of the digital signature in order to control access to the

API...", the Examiner maintains the below-cited grounds of rejection. "Java applets" (as

disclosed within Gibbs) incorporate the functionality associated with a "virtual machine".

Further, Gibbs discloses an "authentication process 'A'" within column 10, lines 14-30

for processing the submitted "unique digital signature 132".

6.   Arguments for independent claims 62, 83, 92, 99, 103, 112, 124, 133, 1443, 152

and 160 were not presented and thus the grounds of rejection set forth for those claims

are hereby maintained.

Claim Objections

7.   Claim 112 is objected to because of the following informalities: spelling error. Claim 112

has the term "identificaters". The Examiner will broadly interpret this term to be "identifiers".

Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section
122(b), by another filed in the United States before the invention by the applicant for

patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 57-71, 74-77, 79-87, 90-97, 99-108, 11-118, 121-127, 130-137, 140-147, 150-156, 159-162 and 165 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,795,919 to Gibbs et al., hereinafter Gibbs.

8.   Regarding claim 57, Gibbs discloses a code signing system for operation in conjunction with a software application having a digital signature and a signature identification, where the digital signature is associated with the signature identification, comprising:

an application platform (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62);

an application programming interface (API) having an associated signature identifier, the API is configured to link the software application with the application platform (column 5, lines 40-51, "an application programming interface ('API') for messaging server 308 is added which provides access to the authenticated message server services");

and a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application if the signature identifier corresponds to the signature identification (column 10, lines 14-30, "Java applets" and lines 35-45).

9.   Regarding claim 58, Gibbs discloses wherein the virtual machine denies the software application access to the API if the digital signature is not authenticated (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

10.   Regarding claim 59, Gibbs discloses, wherein the virtual machine purges the software

application if the digital signature is not authenticated (column 10, lines 63-67, "One response is

to completely disregard the failed unique digital signature 132 and voting responses and delete

them from the unique digital signature system 400").

11.   Regarding claim 60, Gibbs discloses, wherein the code signing system is installed on a

mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-

13).

12.   Regarding claim 61, Gibbs discloses, wherein the digital signature is generated by a

code signing authority (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5,

lines 45-51 and column 8, lines 40-47).

13.   Regarding claim 62, Gibbs discloses a code signing system for operation in conjunction

with a software application having a digital signature and a signature identification where the

digital signature is associated with the signature identification, comprising:

an application platform (column 3, lines 10-18, "an authenticated message server functionally

comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server

316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-

66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62);

a plurality of application programming interfaces (APIs) associated with a signature identifier,

each configured to link the software application with a resource on the application platform

(column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is

added which provides access to the authenticated message server services");

a virtual machine that verifies the authenticity of the digital signature in order to control access

to the APIs by the software application if the signature identification corresponds to the

signature identifier, wherein the virtual machine verifies the authenticity of the digital signature in

order to control access to the plurality of APIs by the software application (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

14.   Regarding claim 63, Gibbs discloses, wherein the plurality of APIs are included in an API library (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 26-30 and column 11, lines 8-12, "add-on software component in servers 424 or 408").

15.   Regarding claim 64, Gibbs discloses, wherein one or more of the plurality of APIs is classified as sensitive and having an associated signature identifier, and wherein the virtual machine uses the digital signature and the signature identification to control access to the sensitive APIs (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature 132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7).

16.   Regarding claim 65, Gibbs discloses, wherein the code signing system operates in conjunction with a plurality of software applications, wherein one or more of the plurality of software applications has a digital signature and a signature identification, and wherein the virtual machine verifies the authenticity of the digital signature of each of the one or more of the plurality of software applications, if the signature identification corresponds to the signature identifier of the respective sensitive APIs, in order to control access to the sensitive APIs by each of the plurality of software applications (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature 132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7).

17.   Regarding claim 66, Gibbs discloses, wherein the resource on the application platform comprises a wireless communication system (Figure 3, element 332 and Figure 4, element 452, column 5, lines 29-33, "a wireless interconnection" and column 7, lines 4-8).

18.   Regarding claim 67, Gibbs discloses, wherein the resource on the application platform comprises a cryptographic module which implements cryptographic algorithms (column 8, lines 11-32 and column 9, lines 36-50).

19.   Regarding claim 68, Gibbs discloses, wherein the resource on the application platform comprises a data store (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62).

20.   Regarding claim 69, Gibbs discloses, wherein the resource on the application platform comprises a user interface (UI) (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62).

        [A "personal computer" contains the means for a user interface.]

21.   Regarding claim 70, Gibbs discloses further comprising:

a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and column 11, lines 8-12, "add-on software component in servers 424 or 408").

22.   Regarding claim 71, Gibbs discloses, wherein at least one of the plurality of API libraries

is classified as sensitive (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines

31-67, "electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7);

wherein access to a sensitive API library requires a digital signature associated with a signature

identification where the signature identification corresponds to a signature identifier associated

with the sensitive API library (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10,

lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature

132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-

7);

wherein the software application includes at least one digital signature and at least one

associated signature identification for accessing sensitive API libraries (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7);

wherein the virtual machine authenticates the software application for accessing the sensitive

API library by verifying a one of the at least one digital signature included in the software

application that has a signature identification corresponding to the signature identifier of the

sensitive API library (column 10, lines 14-30 and 35-45).

23.   Regarding claim 74, Gibbs discloses, wherein the API further comprises:

a description string that is displayed by the mobile device when the software application

attempts to access the API (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

24.   Regarding claim 75, Gibbs discloses, wherein the application platform comprises an operating system (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62).

25.   Regarding claim 76, Gibbs discloses, wherein the application platform comprises one or more core functions of a mobile device (column 6, lines 45-60, column 8, lines 48-55 and column 10, lines 8-13).

26.   Regarding claim 77, Gibbs discloses, wherein the application platform comprises hardware on a mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

27.   Regarding claim 79, Gibbs discloses, wherein the software application is a Java application for a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

28.   Regarding claim 80, Gibbs discloses, wherein the API interfaces with a cryptographic routine on the application platform (column 8, lines 11-32 and column 9, lines 36-50).

29.   Regarding claim 81, Gibbs discloses, wherein the API interfaces with a proprietary data model on the application platform (column 10, lines 31-67, "electronic voting or polling system" and column 11, lines 1-12).

30.   Regarding claim 82, Gibbs discloses, wherein the virtual machine is a virtual machine installed on a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

31.   Regarding claim 83, Gibbs teaches a method of controlling access to sensitive

application programming interfaces on a mobile device, comprising the steps of:

loading a software application on the mobile device that requires access to a sensitive

application programming interface (API) having a signature identifier (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts the remote user for access to

any unique digital signatures stored in a cookie file on laptop 452" and lines 31-67, "electronic

voting or polling system. In such a system, the unique digital signature 132 is allocated by the

authenticated message server 428 at step 704" and column 11, lines 1-7);

determining whether the software application includes a digital signature and a signature

identification (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67,

"electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7);

denying the software application access to the sensitive API if the signature identification does

not correspond with the signature identifier (Figure 6, column 8, lines 56-65 and column 10,

lines 14-30).

32.   Regarding claim 84, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device if the signature

identification does not correspond with the signature identifier (column 10, lines 63-67, "One

response is to completely disregard the failed unique digital signature 132 and voting responses

and delete them from the unique digital signature system 400").

33.   Regarding claim 85, Gibbs teaches, wherein the digital signature and the signature

identification are generated by a code signing authority (column 3, lines 8-18, 25-40 and 64-67,

column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

34.   Regarding claim 86, Gibbs teaches, comprising the additional steps of:

verifying the authenticity of the digital signature where the signature identification

corresponds with the signature identifier (Figure 6, column 8, lines 56-65 and column 10, lines

14-30);

denying the software application access to the sensitive API if the digital signature is not

authenticated (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

35.   Regarding claim 87, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device if the digital signature is not

authenticated (column 10, lines 63-67, "One response is to completely disregard the failed

unique digital signature 132 and voting responses and delete them from the unique digital

signature system 400").

36.   Regarding claim 90, Gibbs teaches, comprising the additional step of:

displaying a description string that notifies a user of the mobile device that the software

application requires access to the sensitive API (column 8, lines 28-39, "a limited character

ASCII set is used since remote users on legacy electronic message and existing telephone

systems can still type the unique digital signature without special software (or hardware)" and

column 10, lines 14-30, "media files are returned, such as Java applets, one or more bundled

HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

37.   Regarding claim 91, Gibbs teaches, comprising the additional step of:

receiving a command from the user granting *or* denying the software application access to the

sensitive API (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

38.   Regarding claim 92, Gibbs teaches a method of controlling access to an application

programming interface (API) having a signature identifier on a mobile device by a software

application created by a software developer, comprising the steps of:

receiving the software application from the software developer (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45);

determining whether the software application satisfies at least one criterion (column 7, lines 23-

38 and column 9, lines 51-58, "other processing/handling of the request for service continues -

depending on the particular application where the unique digital signature is employed");

appending a digital signature and a signature identification to the software application if the

software application satisfies at least one criterion (column 8, lines 11-27, "the digital signature

123 generated at step 516 is converted from a binary value to a value acceptable for electronic

messaging, i.e., ASCII text, by adaptation algorithm 128", column 8, lines 56-65, column 10,

lines 14-30 and column 11, lines 41-43);

verifying the authenticity of the digital signature appended to the software application if the

signature identification corresponds with the signature identifier (Figure 6, column 8, lines 56-65

and column 10, lines 14-30);

providing access to the API to software applications if the digital signature is authenticated

(column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is

added which provides access to the authenticated message server services" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62).

39.   Regarding claim 93, Gibbs teaches, wherein the step of determining whether the

software application satisfies at least one criterion is performed by a code signing authority

(Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

40.   Regarding claim 94, Gibbs teaches, wherein the step of appending the digital signature

and the signature identification to the software application includes generating the digital

signature comprising the steps of:

calculating a hash of the software application (column 3, lines 49-58, "the input to the one-way

hash function" and column 8, lines 11-34, "MD5 function" and "SHA-1 hash function");

applying a signature key to the hash of the software application to generate the digital signature

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50).

41.   Regarding claim 95, Gibbs teaches, wherein the hash of the software application is

calculated using the Secure Hash Algorithm (SHA-1) (column 8, lines 11-27).

42.   Regarding claim 96, Gibbs teaches, wherein the step of verifying the authenticity of the

digital signature comprises the steps of:

providing a corresponding signature key on the mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13);

calculating the hash of the software application on the mobile device to obtain a calculated hash

(column 3, lines 49-58, "the input to the one-way hash function" and column 8, lines 11-34,

"MD5 function" and "SHA-1 hash function");

applying the corresponding signature key to the digital signature to obtain a recovered hash

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50);

authenticating the digital signature by comparing the calculated hash with the recovered hash

(column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital signature 144, is

compared against the adapted digital signature in the incoming unique digital signature").

43.   Regarding claim 97, Gibbs teaches, comprising the further step of denying the software

application access to the API if the digital signature is not authenticated (Figure 6, column 8,

lines 56-65 and column 10, lines 14-30).

44.   Regarding claim 99, Gibbs teaches a method of controlling access to a sensitive

application programming interface (API) having a signature identifier on a mobile device,

comprising the steps of:

registering one or more software developers that are trusted to develop software applications

which access the sensitive API (column 2, lines 28-35, "a value derived from the hash is

concatenated with a service id. In one embodiment, the service id is a local username", column

3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and 40-47 and column 9, lines 41-50);

receiving a hash of a software application (column 2, lines 28-35, column 3, lines 49-67, column

4, lines 1-4, column 8, lines 19-32 and 40-47 and 56-65, column 9, lines 41-50 and column 10,

lines 14-30);

determining whether the hash was sent by a registered software developer (Figure 6, column 8,

lines 56-65, "the service id 104 is tested to verify that it represents a valid local username or

service name" and column 10, lines 14-30);

generating a digital signature using the hash of the software application and a signature

identification corresponding to the signature identifier if the hash was sent by the registered

software developer (column 3, lines 49-58, "the input to the one-way hash function" and column

8, lines 11-34, "MD5 function" and "SHA-1 hash function");

wherein the digital signature and the signature identification are appended to the software

application (column 8, lines 11-27, "the digital signature 123 generated at step 516 is converted

from a binary value to a value acceptable for electronic messaging, i.e., ASCII text, by

adaptation algorithm 128", column 8, lines 56-65, column 10, lines 14-30, "media files are

returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a

RAM file" and lines 49-62 and column 11, lines 41-43);

the mobile device verifies the authenticity of the digital signature in order to control access to the

sensitive API by the software application <u>if</u> the signature identification corresponds with the

signature identifier (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-

67, "electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7).

45.   Regarding claim 100, Gibbs teaches, wherein the step of generating the digital signature

is performed by a code signing authority (column 3, lines 8-18, 25-40 and 64-67, column 4, lines

1-8, column 5, lines 45-51 and column 8, lines 40-47).

46.   Regarding claim 101, Gibbs teaches, wherein the step of generating the digital

signature is performed by applying a signature key to the hash of the software application

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50).

47.   Regarding claim 102, Gibbs teaches, wherein the mobile device verifies the authenticity

of the digital signature by performing the additional steps of:

providing a corresponding signature key on the mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13);

calculating the hash of the software application on the mobile device to obtain a calculated hash

(column 3, lines 49-58, "the input to the one-way hash function" and column 8, lines 11-34,

"MD5 function" and "SHA-1 hash function");

applying the corresponding signature key to the digital signature to obtain a recovered hash

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50);

determining whether the digital signature is authentic by comparing the calculated hash with the

recovered hash (column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital signature

144, is compared against the adapted digital signature in the incoming unique digital signature");

denying the software application access to the sensitive API if the digital signature is not

authenticated (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

48.   Regarding claim 103, Gibbs teaches a method of restricting access to application

programming interfaces on a mobile device, comprising the steps of:

loading a software application having a digital signature and a signature identification on the

mobile device that requires access to one or more application programming interfaces (APIs)

having at least one signature identifier (column 8, lines 11-27, "the digital signature 123

generated at step 516 is converted from a binary value to a value acceptable for electronic

messaging, i.e., ASCII text, by adaptation algorithm 128", column 8, lines 56-65, column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62 and column 11, lines 41-43);

authenticating the digital signature where the signature identification corresponds with the

signature identifier (column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital

signature 144, is compared against the adapted digital signature in the incoming unique digital

signature");

denying the software application access to the one or more APIs where the software application

does not include an authentic digital signature (Figure 6, column 8, lines 56-65 and column 10,

lines 14-30).

49.   Regarding claim 104, Gibbs teaches, wherein the digital signature and signature

identification are associated with a type of mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13).

50.   Regarding claim 105, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device where the software application does not

include an authentic digital signature (column 10, lines 63-67, "One response is to completely

disregard the failed unique digital signature 132 and voting responses and delete them from the

unique digital signature system 400").

51.   Regarding claim 106, Gibbs teaches, wherein:

the software application includes a plurality of digital signatures and signature

identifications (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19

and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and

column 11, lines 8-12, "add-on software component in servers 424 or 408");

the plurality of digital signatures and signature identifications includes digital signatures and

signature identifications respectively associated with different types of mobile devices (column

4, lines 19-42, "log file", "each of the unique digital signatures" and "100 records" and 50-65,

column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines

14-30 and 35-49).

52.   Regarding claim 107, Gibbs teaches, wherein each of the plurality of digital signatures

and associated signature identifications are generated by a respective corresponding code

signing authority (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines

45-51 and column 8, lines 40-47).

53.   Regarding claim 108, Gibbs teaches, wherein the step of determining whether the

software application includes an authentic digital signature comprises the additional steps of:

verifying the authenticity of the digital signature if the signature identification corresponds with

respective ones of the at least one signature identifier (Figure 6, column 8, lines 56-65 and

column 10, lines 14-30).

54.   Regarding claim 111, Gibbs teaches, wherein:

the mobile device includes a plurality of APIs;

at least one of the plurality of APIs is classified as sensitive (column 8, lines 56-67, column 9,

lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In such a

system, the unique digital signature 132 is allocated by the authenticated message server 428

at step 704" and column 11, lines 1-7);

access to any of the plurality of APIs requires an authentic global signature (column 5, lines 40-

51, "an application server interface ('API') for messaging server 308 is added which provides

access to the authenticated message server services" and column 10, lines 14-30, "media files

are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file,

or a RAM file" and lines 49-62);

access to each of the plurality of sensitive APIs requires an authentic global signature and an

authentic digital signature associated with a signature identification (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7);

   the step of determining whether the software application includes an authentic digital

signature and signature identification comprises the steps of:

determining whether the one or more APIs to which the software application requires access

includes a sensitive API (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines

31-67, "electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7);

determining whether the software application includes an authentic global signature (column 5,

lines 40-51, "an application server interface ('API') for messaging server 308 is added which

provides access to the authenticated message server services" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62);

determining whether the software application includes an authentic digital signature and

signature identification if the one or more APIs to which the software application requires access

includes a sensitive API and the software application includes an authentic global signature

(column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is

added which provides access to the authenticated message server services" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62);

the step of denying the software application access to the one or more APIs comprises the

steps of:

denying the software application access to the one or more APIs if the software application does

not include an authentic global signature (Figure 6, column 8, lines 56-65 and column 10, lines

14-30);

denying the software application access to the sensitive API if the one or more APIs to which

the software application requires access includes a sensitive API, the software application

includes an authentic global signature, and the software application does not include an

authentic digital signature and signature identifier required to access the sensitive API (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

55.   Regarding claim 112, Gibbs discloses a code signing system for controlling access to

application programming interfaces (APIs) having signature identificaters by software

applications, the code signing system comprising:

a verification system for executing on a processor and for authenticating digital signatures

included in the respective software applications to access the APIs if the signature identifiers

correspond with the signature identifier of the respective APIs and if a digital signature for a

software application is generated with a signature identification corresponding to a signature

identifier to access at least one API (column 8, lines 56-65, column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming unique

digital signature" and column 10, lines 14-30, "media files are returned, such as Java applets,

one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

a control system for executing on a processor and for allowing access to at least one of the

APIs if the digital signature included in the software application is authenticated by the

verification system (column 5, lines 40-51, "an application server interface ('API') for messaging

server 308 is added which provides access to the authenticated message server services").

56.   Regarding claim 113, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

57.   Regarding claim 114, Gibbs discloses, wherein the virtual machine is a Java virtual

machine installed on a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java applets",

column 10, lines 14-30 and 35-45).

58.   Regarding claim 115, Gibbs discloses, wherein the control system requires one digital

signature and one signature identification for each library of at least one of the APIs (column 4,

lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines

45-60, column 8, lines 2-10, column 10, lines 14-38 and 49-56 and column 11, lines 8-12, "add-

on software component in servers 424 or 408").

59.   Regarding claim 116, Gibbs discloses, wherein the code signing system is installed on a

mobile device and the software application is a Java application for a mobile device (column 6,

lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

60.   Regarding claim 117, Gibbs discloses, wherein the digital signature and the signature

identification of the software application are generated by a code signing authority (column 3,

lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-

47).

61.   Regarding claim 118, Gibbs discloses, wherein the APIs access *at least one of* a

cryptographic module, which implements cryptographic algorithms, a data store, a proprietary

data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

62.   Regarding claim 121, Gibbs discloses, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

63.   Regarding claim 122, Gibbs discloses, wherein the APIs provides access to at least one

of one or more core functions of a mobile device, an operating system, and hardware on a

mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-

13).

64.   Regarding claim 123, Gibbs discloses, wherein verification of a global digital signature

provided <u>to</u> the software application is required for accessing any of the APIs (Figure 6, column

8, lines 56-65 and column 10, lines 14-30).

65.   Regarding claim 124, Gibbs teaches a method of controlling access to application

programming interfaces (APIs) having signature identifiers by software applications, the method

comprising:

executing instructions on a processor for authenticating digital signatures provided to the

respective software applications to access the APIs if the signature identifications correspond

with the signature identifiers of the respective APIs and if a digital signature for a software

application is generated with a signature identification corresponding to a signature identifier to

access at least one API (column 8, lines 56-65, column 9, lines 36-58, "an adapted digital

signature 144, is compared against the adapted digital signature in the incoming unique digital

signature" and column 10, lines 14-30, "media files are returned, such as Java applets, one or

more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

executing instructions on a processor for allowing access to at least one of the APIs if the digital

signature provided to the software application is authenticated (column 5, lines 40-51, "an

application server interface ('API') for messaging server 308 is added which provides access to

the authenticated message server services").

66.   Regarding claim 125, Gibbs teaches, wherein one digital signature and one signature

identification are provided to the software application access a library of at least one of the APIs

(column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56,

column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and column 11,

lines 8-12, "add-on software component in servers 424 or 408").

67.   Regarding claim 126, Gibbs teaches, wherein the digital signature and the signature

identification of the software application are generated by a code signing authority (column 3,

lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-

47).

68.   Regarding claim 127, Gibbs teaches, wherein the APIs access *at least one of* a

cryptographic module that implements cryptographic algorithms, a data store, a proprietary data

model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

69.   Regarding claim 130, Gibbs teaches, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

70.   Regarding claim 131, Gibbs teaches, wherein the APIs provides access to at least one

of one or more core functions of a mobile device, an operating system, and hardware on a

mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-

13).

71.   Regarding claim 132, Gibbs teaches, wherein verification of a global digital signature

provided <u>to</u> the software application is required for accessing any of the APIs (Figure 6, column

8, lines 56-65 and column 10, lines 14-30).

72.   Regarding claim 133, Gibbs discloses a system for controlling access by software

applications to application programming interfaces (APIs) having at least one signature identifier

on a subset of a plurality of mobile devices, the management system comprising:

a code signing authority for <u>associating</u> digital signatures and signature identifications <u>with</u>

software applications that require access to at least one of the APIs with a signature identifier on

the subset of the plurality of mobile devices, <u>if</u> a digital signature for a software application is

generated with a signature identification corresponding to a signature identifier, and the

signature identifications provided to the software applications comprise those signature

identifications that correspond to the signature identifiers that are on the subset of the plurality

of mobile devices (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-

13);

wherein each mobile device of the subset of the plurality of mobile devices comprises a

verification system for authenticating digital signatures provided <u>to</u> the respective software

applications to access respective APIs <u>if</u>the digital identifications correspond to the digital

identifiers of the respective APIs (column 8, lines 56-65, column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming unique

digital signature" and column 10, lines 14-30 and 49-62);

a control system for allowing the respective software applications to access at least one of the

APIs <u>if</u> the digital signatures provided <u>to</u> the respective software applications are authenticated

by the verification system (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10,

lines 8-13, "prompts the remote user for access to any unique digital signatures stored in a

cookie file on laptop 452" and lines 31-67, "electronic voting or polling system. In such a

system, the unique digital signature 132 is allocated by the authenticated message server 428

at step 704" and column 11, lines 1-7).

73.   Regarding claim 134, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

74.   Regarding claim 135, Gibbs discloses, wherein the virtual machine is a Java virtual

machine and the software applications are Java applications (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

75.   Regarding claim 136, Gibbs discloses, wherein the control system requires one digital

signature and one signature identification for each library of at least one of the APIs (column 4,

lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines

45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12, "add-on software

component in servers 424 or 408").

76.   Regarding claim 137, Gibbs discloses, wherein the APIs access at least one of a

cryptographic module, which implements cryptographic algorithms, a data store, a proprietary

data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

77.   Regarding claim 140, Gibbs discloses, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

78.   Regarding claim 141, Gibbs discloses, wherein the subset of the plurality of mobile devices comprises mobile devices under the control of at least one of a corporation and a carrier (column 2, lines 8-16, "subscribers of a particular service" and column 5, lines 52-65).

79.   Regarding claim 142, Gibbs discloses, wherein a global digital signature provided by the software application has to be authenticated before the software application is allowed access to any of the APIs on a mobile device of the subset of the plurality of mobile devices (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

80.   Regarding claim 143, Gibbs teaches a method of controlling access by software applications to application programming interfaces (APIs) having at least one signature identifier on a subset of a plurality of mobile devices, the method comprising:

generating digital signatures for software applications with signature identifications corresponding to respective signature identifiers of the APIs (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 41-51 and column 8, lines 40-47);

and providing the digital signatures and the signature identifications to software applications that require access to at least one of the APIs on the subset of the plurality of mobile devices, if the signature identifications provided to the software applications comprise those signature identifications that correspond to the signature identifiers that are on the subset of the plurality of mobile devices (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and column 11, lines 8-12, "add-on software component in servers 424 or 408");

wherein each mobile device of the subset of the plurality of mobile devices comprises a verification system for authenticating digital signatures provided to the respective software

applications to access respective APIs if the digital identifications correspond to the digital

identifiers of the respective APIs (column 8, lines 56-65, column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming unique

digital signature" and column 10, lines 14-30, "media files are returned, such as Java applets,

one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

a control system for allowing the software application to access at least one of the APIs if the

digital signature provided to the software application is authenticated by the verification system

(column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts the

remote user for access to any unique digital signatures stored in a cookie file on laptop 452" and

lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature

132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-

7).

81.   Regarding claim 144, Gibbs teaches, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

82.   Regarding claim 145, Gibbs teaches, wherein the virtual machine is a Java virtual

machine and the software applications are Java applications (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

83.   Regarding claim 146, Gibbs teaches, wherein the control system requires one digital

signature and one signature identification for each library of at least one of the APIs (column 4,

lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines

45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12, "add-on software

component in servers 424 or 408").

84.   Regarding claim 147, Gibbs teaches, wherein the APIs access at least one of a

cryptographic module, which implements cryptographic algorithms, a data store, a proprietary

data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

85.   Regarding claim 150, Gibbs teaches, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

86.   Regarding claim 151, Gibbs teaches, wherein the subset of the plurality of mobile

devices comprises mobile devices under the control of at least one of a corporation and a

carrier (column 2, lines 8-16, "subscribers of a particular service" and column 5, lines 52-65).

87.   Regarding claim 152, Gibbs teaches a mobile device comprising:

an application platform having application programming interfaces (APIs) (column 6, lines 45-

60, "laptop", column 8, lines 48-55 and column 10, lines 8-13);

a verification system for authenticating digital signatures and signature identifications provided

to the respective software applications to access the APIs (column 8, lines 56-65, column 9,

lines 36-58, "an adapted digital signature 144, is compared against the adapted digital signature

in the incoming unique digital signature" and column 10, lines 14-30, "media files are returned,

such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file"

and lines 49-62);

a control system for allowing a software application to access at least one of the APIs if a digital

signature provided to the software application is authenticated by the verification system

(column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts the

remote user for access to any unique digital signatures stored in a cookie file on laptop 452" and

lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature

132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-

7);

wherein a code signing authority provides digital signatures and signature identifications to

software applications that require access to at least one of the APIs such that the digital

signature for the software application is generated according to a signature scheme of a

signature identification, and wherein the signature identifications provided to the software

applications comprise those signature identifications that are authorized to allow access on a

subset of a plurality of mobile devices (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-

8, column 5, lines 45-51 and column 8, lines 40-47).

88.   Regarding claim 153, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

89.   Regarding claim 154, Gibbs discloses, wherein the virtual machine is a Java virtual

machine and the software application is a Java application (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

90.   Regarding claim 155, Gibbs discloses, wherein the control system requires one digital

signature and one signature identification for each library of at least one of the APIs (column 4,

lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines

45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12, "add-on software

component in servers 424 or 408").

91.   Regarding claim 156, Gibbs discloses, wherein the APIs of the application platform

access at least one of a cryptographic module, which implements cryptographic algorithms, a

data store, a proprietary data model, and a user interface (UI) (column 3, lines 10-18, "an

authenticated message server functionally comprises a digital service engine 120", column 5,

lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and

lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer",

column 8, lines 11-32, column 9, lines 36-50 and column 10, lines 49-62).

92.   Regarding claim 159, Gibbs discloses, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

93.   Regarding claim 160, Gibbs teaches a method of controlling access to application

programming interfaces (APIs) of an application platform of a mobile device, the method

comprising:

receiving digital signatures and signature identifications from software applications that require

access to the APIs; authenticating the digital signatures and the signature identifications

(column 8, lines 56-65, column 9, lines 36-58, "an adapted digital signature 144, is compared

against the adapted digital signature in the incoming unique digital signature" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62);

allowing a software application to access at least one of the APIs if a digital signature provided

to the software application is authenticated (column 8, lines 56-67, column 9, lines 1-5 and 45-

50, column 10, lines 8-13, "prompts the remote user for access to any unique digital signatures

stored in a cookie file on laptop 452" and lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7);

wherein a code signing authority provides the digital signatures and the signature identifications

to the software applications that require access to at least one of the APIs such that the digital

signature for the software application is generated according to a signature scheme of a

signature identification, and wherein the signature identifications provided to the software

applications comprise those signature identifications that are authorized to allow access on a

subset of a plurality of mobile devices (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-

8, column 5, lines 45-51 and column 8, lines 40-47).

94.   Regarding claim 161, Gibbs teaches, wherein one digital signature and one signature

identification is required for accessing each library of at least one of the APIs (column 4, lines

19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines 45-

60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12, "add-on software

component in servers 424 or 408").

95.   Regarding claim 162, Gibbs teaches, wherein the APIs of the application platform

access at least one of a cryptographic module, which implements cryptographic algorithms, a

data store, a proprietary data model, and a user interface (UI) (column 3, lines 10-18, "an

authenticated message server functionally comprises a digital service engine 120", column 5,

lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and

lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer",

column 8, lines 11-32, column 9, lines 36-50 and column 10, lines 49-62).

96.   Regarding claim 165, Gibbs teaches, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to

access said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is

used since remote users on legacy electronic message and existing telephone systems can still

type the unique digital signature without special software (or hardware)" and column 10, lines

14-30, "media files are returned, such as Java applets, one or more bundled HTML files, an

MPEG file, a WAV file, or a RAM file" and lines 49-62).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject
> matter sought to be patented and the prior art are such that the subject matter as a
> whole would have been obvious at the time the invention was made to a person having
> ordinary skill in the art to which said subject matter pertains. Patentability shall not be
> negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459

(1966), that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating obviousness
       or nonobviousness.

97.   Claim 78 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gibbs as

applied to claim 57 above, and further in view of United States Patent No. 6,584,376 to Van

Kommer, hereinafter Van Kommer.

98.   Gibbs significantly discloses the claimed invention, as cited above. However, Gibbs

does not substantially disclose the claim language of claim 78 pertaining to "wherein the

hardware comprises a subscriber identity module (SIM) card". Van Kommer discloses this, as

cited below.

99.   Regarding claim 78, Van Kommer discloses, wherein the hardware comprises a

subscriber identity module (SIM) card (column 3, lines 22-33, "mobile phone comprises

preferably a subscriber identification module 300, for example a removable SIM card, which

enables the mobile robot to be identified within the mobile telecommunications network 2" and

column 5, lines 31-36).

100.   The motivation to combine would be to provide a means for "the mobile robot to be

identified within the mobile telecommunications network" (*Van Kommer* – column 3, lines 24-

27).

101.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to incorporate the teachings of Van Kommer with the teachings of Gibbs in

order "to verify the identity and privileges of the distant operator 1, for example a mobile

subscriber" (*Van Kommer* - column 6, lines 56-64).

102.   Claims 72, 73, 88, 89, 98, 109, 110, 119, 120, 128, 129, 138, 139, 148, 149, 157, 158,

163 and 164 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gibbs as applied

to claims 57, 83, 92, 103, 112, 124, 133, 143, 152 and 160 above, and further in view of United

States Patent No. 6,587,837 to Spagna et al., hereinafter Spagna.

103.   Gibbs significantly discloses the claimed invention, as cited above. However, Gibbs

does not substantially disclose the claim language pertaining to the "public signature key" and

"private signature key" as found within the following claims. Spagna discloses this claim

language, as cited below.

104.   Based upon the similarities of the claim language between the following claims, the

following motivation and obviousness to combine is applicable to each of the subsequent

claims.

105.   The motivation to combine would be that "the issuer of SC(s) protects the integrity of

SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

106.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to incorporate the teachings of Spagna with the teachings of Gibbs so that

"it is easy to calculate the digest for an input message, but it is very difficult (computationally

infeasible) to generate "he input message from its digest" (*Spagna* – column 17, lines 25-31).

107.   Regarding claim 72, Spagna discloses, wherein the digital signature is generated using

a private signature key, and the virtual machine uses a public signature key to verify the

authenticity of the digital signature (column 16, lines 46-53, "Public key algorithms are also used

to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14

and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the

signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and

column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and

decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity

included if signed").

108.   Regarding claim 73, Spagna discloses, wherein:

the digital signature is generated by applying the private signature key to a hash of the software

application (column 17, lines 25-33, column 27, lines 41-51);

the virtual machine verifies the authenticity of the digital signature by generating a hash of the

software application to obtain a generated hash, applying the public signature key to the digital

signature to obtain a recovered hash, and comparing the generated hash with the recovered

hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36,

lines 9-19 and 43-49).

109.   Regarding claim 88, Spagna teaches, wherein the digital signature is generated by

applying a private signature key to a hash of the software application (column 16, lines 46-53,

"Public key algorithms are also used to generate digital signatures. The private key is used for

that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which

private key was used to create the signature thus in EU is the End-User(s) digital signature",

column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed"),

and wherein the step of verifying the authenticity of the digital signature is performed by a

method comprising the steps of:

storing a public signature key that corresponds to the private signature key on the mobile device

(column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures.

The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-

11, "initials indicate which private key was used to create the signature thus in EU is the End-

User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To

validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of

the signature itself using the Public Key 661 of the signing entity included if signed");

generating a hash of the software application to obtain a generated hash (column 17, lines 25-33, column 27, lines 41-51);

applying the public signature key to the digital signature to obtain a recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49);

comparing the generated hash with the recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

110.   Regarding claim 89, Spagna teaches, wherein the digital signature is generated by calculating a hash of the software application and applying the private signature key (column 17, lines 25-33, column 27, lines 41-51).

111.   Regarding claim 98, Spagna teaches, wherein the signature key is a private signature key and the corresponding signature key is a public signature key (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

112.   Regarding claim 109, Spagna teaches, wherein each of the plurality of digital signatures and signature identifications are generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the

signature thus in EU is the End-User(s) digital signature", column 18, lines 46-60, column 19,

lines 19-30, column 27, lines 41-51, column 36, lines 9-19 and 43-49 and column 46, lines 40-

58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents

631 of the signature itself using the Public Key 661 of the signing entity included if signed").

113.   Regarding claim 110, Spagna teaches, wherein the step of authenticating the digital

signature if the signature identification corresponds with the signature identifier comprises the

steps of:

verifying that the signature identification corresponds with the signature identifier; wherein

authenticating the digital signature if signature identification corresponds with the signature

identifier comprising the steps of:

storing a public signature key on a mobile device that corresponds to the private signature key

associated with the code signing authority which generates the digital signature (column 16,

lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key

is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials

indicate which private key was used to create the signature thus in EU is the End-User(s) digital

signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed");

generating a hash of the software application to obtain a generated hash (column 17, lines 25-

33, column 27, lines 41-51);

applying the public signature key to the digital signature to obtain a recovered hash (column 18,

lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-

49);

and comparing the generated hash with the recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

114.   Regarding claim 119, Spagna discloses, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and the verification system uses a public signature key to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

115.   Regarding claims 120, 139, 158 and 164, Spagna discloses wherein:
the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme (column 17, lines 25-33, column 27, lines 41-51);
the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

116.   Regarding claim 128, Spagna teaches, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and a public signature key is used to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which

private key was used to create the signature thus in EU is the End-User(s) digital signature",

column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed").

117.   Regarding claim 129, Spagna teaches, wherein:

the digital signature is generated by applying the private signature key to a hash of the software

application under the signature scheme (column 17, lines 25-33, column 27, lines 41-51);

the digital signature is authenticated by generating a hash of the software application to obtain a

generated hash, applying the public signature key to the digital signature to obtain a recovered

hash, and verifying that the generated hash with the recovered hash are the same (column 18,

lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-

49).

118.   Regarding claim 138, Spagna discloses, wherein the digital signature is generated using

a private signature key under a signature scheme associated with the signature identification,

and the verification system uses a public signature key to authenticate the digital signature

(column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures.

The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-

11, "initials indicate which private key was used to create the signature thus in EU is the End-

User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To

validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of

the signature itself using the Public Key 661 of the signing entity included if signed").

119.   Regarding claim 148, Spagna teaches, wherein at least one of the digital signatures is

generated using a private signature key under a signature scheme associated with a signature

identification, and the verification system uses a public signature keys to authenticate said at

least one of the digital signatures (column 16, lines 46-53, "Public key algorithms are also used

to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14

and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the

signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and

column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and

decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity

included if signed").

120.   Regarding claim 149, Spagna teaches, wherein:

at least one of the digital signatures is generated by applying the private signature key to a hash

of a software application under the signature scheme (column 17, lines 25-33, column 27, lines

41-51);

the verification system authenticates said at least one of the digital signatures by generating a

hash of the software application to obtain a generated hash, applying the public signature key to

said at least one of the digital signatures to obtain a recovered hash, and verifying that the

generated hash with the recovered hash are the same (column 18, lines 46-60, column 19, lines

19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

121.   Regarding claim 157, Spagna discloses, wherein the digital signature is generated using

a private signature key under the signature scheme, and the verification system uses a public

signature key to authenticate the digital signature (column 16, lines 46-53, "Public key

algorithms are also used to generate digital signatures. The private key is used for that

purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which

private key was used to create the signature thus in EU is the End-User(s) digital signature",

column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed").

122.   Regarding claim 163, Spagna teaches, wherein the digital signature is generated using a

private signature key under the signature scheme, and a public signature key is used to

authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used

to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14

and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the

signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and

column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and

decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity

included if signed").

<div align="center">***Conclusion***</div>

123.   Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

124.   A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the date of this final action.

125.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

126.   The following United States Patents and Patent Application Publication are cited to

further show the state of the art with respect to ensuring the security of data, such as:

United States Patent No. 6,574,609 to Downs, et al., which is cited to show a secure

electronic content management system.

United States Patent No. 6,324,650 to Ogilvie, which is cited to show message content

protection and conditional disclosure.

United States Patent No. 6,795,923 to Stern, et al., which is cited to show a mechanism

for embedding network based control systems in a local network interface device.

United States Patent No. 7,243,236 to Silbert, which is cited to show systems and

methods for using cryptography to protect secure and insecure computing environments.

United States Patent Application Publication No. US 2001/0044901 to Grawrock, which

is cited to show a bubble-protected system for automatic decryption of file data on a per-use

basis and automatic re-encryption.

127.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to JEREMIAH AVERY whose telephone number is

(571)272-8627. The examiner can normally be reached on Monday thru Friday 8:30am-

5pm.

128.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, William Korzuch can be reached on (571) 272-7589. The fax phone number

for the organization where this application or proceeding is assigned is 571-273-8300.

129.   Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jeremiah Avery/
Examiner, Art Unit 2431
/Syed   Zia/
Primary Examiner, Art Unit 2431

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Notice of References Cited		10/381,219	YACH ET AL.
		Examiner	Art Unit	
		JEREMIAH AVERY	2431	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,795,919	09-2004	Gibbs et al.	713/170
*	B	US-6,587,837	07-2003	Spagna et al.	705/26
*	C	US-6,574,609	06-2003	Downs et al.	705/50
*	D	US-6,324,650	11-2001	Ogilvie, John W.L.	726/2
*	E	US-6,795,923	09-2004	Stern et al.	726/12
*	F	US-7,243,236	07-2007	Sibert, W. Olin	713/179
*	G	US-6,584,376	06-2003	Van Kommer, Robert	700/245
*	H	US-2001/0044901	11-2001	GRAWROCK, DAVID	713/189
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

U.S. Patent and Trademark Office
PTO-892 (Rev. 01-2001)   **Notice of References Cited**   Part of Paper No. 20091013

Search Notes	**Application/Control No.**	**Applicant(s)/Patent Under Reexamination**
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

SEARCHED			
Class	**Subclass**	**Date**	**Examiner**
none	none	10/20/2009	JLA

SEARCH NOTES		
Search Notes	**Date**	**Examiner**
Updated EAST Search	10/20/2009	JLA

INTERFERENCE SEARCH			
Class	**Subclass**	**Date**	**Examiner**
none	none	10/20/2009	JLA

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE								
Final	Original	10/20/2009	10/20/2009							
	1	-								
	2	-								
	3	-								
	4	-								
	5	-								
	6	-								
	7	-								
	8	-								
	9	-								
	10	-								
	11	-								
	12	-								
	13	-								
	14	-								
	15	-								
	16	-								
	17	-								
	18	-								
	19	-								
	20	-								
	21	-								
	22	-								
	23	-								
	24	-								
	25	-								
	26	-								
	27	-								
	28	-								
	29	-								
	30	-								
	31	-								
	32	-								
	33	-								
	34	-								
	35	-								
	36	-								

	Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
		10381219	YACH ET AL.
		Examiner	**Art Unit**
		JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE								
Final	Original	10/20/2009	10/20/2009							
	37	-								
	38	-								
	39	-								
	40	-								
	41	-								
	42	-								
	43	-								
	44	-								
	45	-								
	46	-								
	47	-								
	48	-								
	49	-								
	50	-								
	51	-								
	52	-								
	53	-								
	54	-								
	55	-								
	56	-								
	57	✓								
	58	✓								
	59	✓								
	60	✓								
	61	✓								
	62	✓								
	63	✓								
	64	✓								
	65	✓								
	66	✓								
	67	✓								
	68	✓								
	69	✓								
	70	✓								
	71	✓								
	72	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE									
Final	Original	10/20/2009	10/20/2009								
	73	✓									
	74	✓									
	75	✓									
	76	✓									
	77	✓									
	78	✓									
	79	✓									
	80	✓									
	81	✓									
	82	✓									
	83	✓									
	84	✓									
	85	✓									
	86	✓									
	87	✓									
	88	✓									
	89	✓									
	90	✓									
	91	✓									
	92	✓									
	93	✓									
	94	✓									
	95	✓									
	96	✓									
	97	✓									
	98	✓									
	99	✓									
	100	✓									
	101	✓									
	102	✓									
	103	✓									
	104	✓									
	105	✓									
	106	✓									
	107	✓									
	108	✓									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE								
Final	Original	10/20/2009	10/20/2009							
	109	✓								
	110	✓								
	111	✓								
	112	✓	O							
	113	✓								
	114	✓								
	115	✓								
	116	✓								
	117	✓								
	118	✓								
	119	✓								
	120	✓								
	121	✓								
	122	✓								
	123	✓								
	124	✓								
	125	✓								
	126	✓								
	127	✓								
	128	✓								
	129	✓								
	130	✓								
	131	✓								
	132	✓								
	133	✓								
	134	✓								
	135	✓								
	136	✓								
	137	✓								
	138	✓								
	139	✓								
	140	✓								
	141	✓								
	142	✓								
	143	✓								
	144	✓								

U.S. Patent and Trademark Office   Part of Paper No. : 20091013

Page 871 of 1415

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant   ☐ CPA   ☐ T.D.   ☐ R.1.47

CLAIM		DATE								
Final	Original	10/20/2009	10/20/2009							
	145	✓								
	146	✓								
	147	✓								
	148	✓								
	149	✓								
	150	✓								
	151	✓								
	152	✓								
	153	✓								
	154	✓								
	155	✓								
	156	✓								
	157	✓								
	158	✓								
	159	✓								
	160	✓								
	161	✓								
	162	✓								
	163	✓								
	164	✓								
	165	✓								

Receipt date: 10/01/2009
Doc description: Information Disclosure Statement (IDS) Filed

10381219 PTO/SB/08a (06-04)
Approved for use through 06/30/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10381219
	Filing Date	2003-03-20
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
	Attorney Docket Number	555255-012423

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5625690	A	1997-04-29	Michel et al.	
	2	6067582	A	2000-05-23	Smith et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

EFS Web 2.1.4

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

Page 873 of 1415

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	10381219	10381219 - GAU: 2431
Filing Date	2003-03-20	
First Named Inventor	David P. Yach	
Art Unit	2431	
Examiner Name	Jeremiah L. Avery	
Attorney Docket Number	555255-012423	

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T⁵
	1	European Search Report issued on May 15, 2009 in connection with European Patent Application No. 05024662.8.	☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature	/Jeremiah Avery/	Date Considered	10/20/2009

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L8	757	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L9	130	L8 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L10	72	L9 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L11	65	L10 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L12	64	L11 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L13	64	L12 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L14	48	L13 and (digital near signature) and ((authentic$ or verify$ or verification or valid$) near signature)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29
L15	6	L14 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:29

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (1 of 6)10/20/2009 6:42:37 PM

Page 875 of 1415

L16	11	L14 and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) near (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/10/20 18:37
L17	21	l14 and (virtual near machine)	US-PGPUB; USPAT	OR	ON	2009/10/20 18:38
S1	8	(digital near signature) and (authentic$ or verify$ or verificat$) and (virtual near machine) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:23
S2	8	S1 and (portab$ or mobile or handheld or laptop or pda or cell or cellular)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S3	4	S2 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S4	35	(digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:28
S5	737	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (2 of 6)10/20/2009 6:42:37 PM

Page 876 of 1415

S6	41	S5 and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5 or revok$ or revocat$) same (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33
S7	30	S6 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:37
S8	30	S7 and access$	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S9	30	S8 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S10	2	S9 and ((secure near hash near algorithm) or SHA?1)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:39
S11	1	S10 and public and private	US-PGPUB; USPAT	OR	ON	2009/02/20 11:40
S12	31	S6 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:46
S13	31	S12 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:47
S14	31	S13 and ((public or private) same key)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:48
S15	0	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) same (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (3 of 6)10/20/2009 6:42:37 PM

Page 877 of 1415

S16	30	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) and (((secure near hash near algorithm) or (SHA1 of SHA? 1)) or (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53
S17	28	S16 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:07
S18	118	S5 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S19	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S20	56	S19 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:26
S21	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S22	56	S21 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S23	40	S22 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:28
S24	55	S22 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:32

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (4 of 6)10/20/2009 6:42:37 PM

Page 878 of 1415

S25	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S26	119	S25 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S27	62	S26 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S28	16	S27 and (SIM or (subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S29	30	S25 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda) and (SIM or (Subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:14
S30	16	S29 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:24
S31	9	S28 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:28
S32	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51

file:///C|/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (5 of 6)10/20/2009 6:42:37 PM

Page 879 of 1415

S33	119	S32 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov$5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S34	62	S33 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S35	57	S34 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S36	56	S35 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S37	56	S36 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S38	36	S37 and (digital near signature) and ((authentic$ or verify$ or verification) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S39	0	S38 and (signature near (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S40	3	S38 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:53
S41	40	S37 and (digital near signature) and ((authentic$ or verify$ or verification or valid$) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57
S42	6	S41 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57

10/20/2009 6:42:24 PM
C:\ Documents and Settings\ javery\ My Documents\ EAST\ Workspaces\ 10381219.wsp

file:///C\/Documents%20and%20Settings/javery/My%20Do...219/EASTSearchHistory.10381219_AccessibleVersion.htm (6 of 6)10/20/2009 6:42:37 PM

Page 880 of 1415

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10381219
	Filing Date	2003-03-20
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
	Attorney Docket Number	555255-012423

			U.S.PATENTS				
Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5625690	A	1997-04-29	Michel et al.	
	2	6067582	A	2000-05-23	Smith et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

			U.S.PATENT APPLICATION PUBLICATIONS				
Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

				FOREIGN PATENT DOCUMENTS				
Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

<table>
<tr><td rowspan="2"></td><td colspan="2">INFORMATION DISCLOSURE</td><td>Application Number</td><td>10381219</td></tr>
</table>

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99**)	Application Number	10381219
	Filing Date	2003-03-20
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
	Attorney Docket Number	555255-012423

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	European Search Report issued on May 15, 2009 in connection with European Patent Application No. 05024662.8.	☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10381219
	Filing Date	2003-03-20
	First Named Inventor	David P. Yach
	Art Unit	2431
	Examiner Name	Jeremiah L. Avery
	Attorney Docket Number	555255-012423

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).
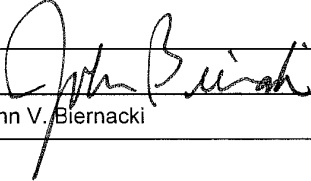
OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

| Signature | | Date (YYYY-MM-DD) | October 1, 2009 |
| Name/Print | John V. Biernacki | Registration Number | 40,511 |

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Electronic Patent Application Fee Transmittal

Application Number:	10381219
Filing Date:	20-Mar-2003
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Filer:	Stephen D. Scanlon/John V. Biernacki
Attorney Docket Number:	555255012423

Filed as Large Entity

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
Total in USD ($)				**180**

Electronic Acknowledgement Receipt

EFS ID:	6185757
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Customer Number:	89441
Filer:	Stephen D. Scanlon/John V. Biernacki
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	555255012423
Receipt Date:	01-OCT-2009
Filing Date:	20-MAR-2003
Time Stamp:	14:50:20
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$ 180
RAM confirmation Number	1020
Deposit Account	501432
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	
Charge any Additional Fees required under 37 C.F.R. 1.492 (National application filing, search, and examination fees)	
Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

| | Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges) | | | | |

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed (SB/08)	DOC056.pdf	137900 76c60e06f6246082729021deb6ad4df04b8 b0421	no	3

Warnings:

Information:

This is not an USPTO supplied IDS fillable form

| 2 | NPL Documents | DOC057.pdf | 103290

eb51e312805c785b1f22afaff13087ec98d1 e175 | no | 4 |

Warnings:

Information:

| 3 | Fee Worksheet (PTO-875) | fee-info.pdf | 30280

9b9d5de2eeba09c8a28f72f742aea1e0083a 439a | no | 2 |

Warnings:

Information:

| | | Total Files Size (in bytes): | 271470 | | |

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

CLAIMS ONLY		Application Number		Filing Date	
		10-381219			
		Applicant(s)			

* May be used for additional claims or amendments

CLAIMS	AS FILED		AFTER FIRST AMENDMENT		AFTER SECOND AMENDMENT				*		*		*	
	Indep	Depend	Indep	Depend	Indep	Depend			Indep	Depend	Indep	Depend	Indep	Depend
1								51						
2								52						
3								53						
4								54						
5								55						
6								56						
7								57						
8								58						
9								59						
10								60						
11								61						
12								62						
13								63						
14								64						
15								65						
16								66						
17								67						
18								68						
19								69						
20								70						
21								71						
22								72						
23								73						
24								74						
25								75						
26								76						
27								77						
28								78						
29								79						
30								80						
31								81						
32								82						
33								83						
34								84						
35								85						
36								86						
37								87						
38								88						
39								89						
40								90						
41								91						
42								92						
43								93						
44								94						
45								95						
46								96						
47								97						
48								98						
49								99						
50								100						
Total Indep								Total Indep	5					
Total Depend								Total Depend	39					
Total Claims								Total Claims	44					

CLAIMS ONLY

Application Number	Filing Date
10-381219	

Applicant(s)

* May be used for additional claims or amendments

CLAIMS	AS FILED		AFTER FIRST AMENDMENT		AFTER SECOND AMENDMENT			*		*		*	
	Indep	Depend	Indep	Depend	Indep	Depend		Indep	Depend	Indep	Depend	Indep	Depend
1		1					51		1				
2		1					52	1					
3	1						53		1				
4		1					54		1				
5		1					55		1				
6		1					56		1				
7		1					57		1				
8		1					58		1				
9		1					59		1				
10		1					60	1					
11		1					61		1				
12	1						62		1				
13		1					63		1				
14		1					64		1				
15		1					65	1					
16		1					66		1				
17		1					67						
18		1					68						
19		1					69						
20		1					70						
21		1					71						
22		1					72						
23		1					73						
24	1						74						
25		1					75						
26		1					76						
27		1					77						
28		1					78						
29		1					79						
30		1					80						
31		1					81						
32		1					82						
33	1						83						
34		1					84						
35		1					85						
36		1					86						
37		1					87						
38		1					88						
39		1					89						
40		1					90						
41		1					91						
42		1					92						
43	1						93						
44		1					94						
45		1					95						
46		1					96						
47		1					97						
48		1					98						
49		1					99						
50		1					100						
Total Indep	5						Total Indep	2					
Total Depend	45						Total Depend	13					
Total Claims	50						Total Claims	15					

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/381,219		2431	26M1

OC00000037317230

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 89441 on 08/11/2009
- Correspondence Address
- Maintenance Fee Address
- Power of Attorney Address

The address of record for Customer Number 89441 is:

89441
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	David P. Yach; Michael S. Brown; Herbert A. Little
Internat'l. Appl'n. No.	:	PCT/CA01/01344
Internat'l. Filing Date	:	09/20/2001
U.S. Serial No.	:	10/381,219
U.S. Filing Date	:	03/20/2003
Title	:	Software Code Signing System And Method
Art Unit	:	2131
Examiner	:	J. Avery
Docket No.	:	555255012423

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSIVE AMENDMENT

Please amend this application as follows and consider the following remarks. Any fees due should be charged to Jones Day Deposit Account No. 501432, ref: 555255-012423.

The Claims

1-56. (Cancelled)

57. (Currently Amended) A code signing system for operation in conjunction with a software application having a digital signature and a signature identification, wherein the digital signature is associated with the signature identification, comprising:

an application platform;

an application programming interface (API) having an associated signature identifier, the API is configured to link the software application with the application platform; and

a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application ~~where~~ if the signature identifier corresponds to the signature identification.

58. (Previously Presented) The code signing system of claim 57, wherein the virtual machine denies the software application access to the API if the digital signature is not authenticated.

59. (Previously Presented) The code signing system of claim 57, wherein the virtual machine purges the software application if the digital signature is not authenticated.

60. (Previously Presented) The code signing system of claim 57, wherein the code signing system is installed on a mobile device.

61. (Previously Presented) The code signing system of claim 57, wherein the digital signature is generated by a code signing authority.

62. (Currently Amended) A code signing system for operation in conjunction with a software application having a digital signature and a signature identification ~~where~~ if the digital signature is associated with the signature identification, comprising:

an application platform;

a plurality of application programming interfaces (APIs) associated with a signature identifier, each configured to link the software application with a resource on the application platform; and

a virtual machine that verifies the authenticity of the digital signature in order to control access to the APIs by the software application ~~where~~ if the signature identification corresponds to the signature identifier,

wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

63. (Previously Presented) The code signing system of claim 62, wherein the plurality of APIs are included in an API library.

64. (Previously Presented) The code signing system of claim 62, wherein one or more of the plurality of APIs is classified as sensitive and having an associated signature identifier, and wherein the virtual machine uses the digital signature and the signature identification to control access to the sensitive APIs.

65.   (Currently Amended) The code signing system of claim 64, wherein the code signing system operates in conjunction with a plurality of software applications, wherein one or more of the plurality of software applications has a digital signature and a signature identification, and wherein the virtual machine verifies the authenticity of the digital signature of each of the one or more of the plurality of software applications, ~~where~~ if the signature identification corresponds to the signature identifier of the respective sensitive APIs, in order to control access to the sensitive APIs by each of the plurality of software applications.

66.   (Previously Presented) The code signing system of claim 62, wherein the resource on the application platform comprises a wireless communication system.

67.   (Previously Presented) The code signing system of claim 62, wherein the resource on the application platform comprises a cryptographic module which implements cryptographic algorithms.

68.   (Previously Presented) The code signing system of claim 62, wherein the resource on the application platform comprises a data store.

69.   (Previously Presented) The code signing system of claim 62, wherein the resource on the application platform comprises a user interface (UI).

70.   (Previously Presented) The code signing system of claim 57, further comprising:

a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

71.   (Currently Amended) The code signing system of claim 70, wherein at least one of the plurality of API libraries is classified as sensitive;

wherein access to a sensitive API library requires a digital signature associated with a signature identification ~~where~~ if the signature identification corresponds to a signature identifier associated with the sensitive API library;

wherein the software application includes at least one digital signature and at least one associated signature identification for accessing sensitive API libraries; and

wherein the virtual machine authenticates the software application for accessing the sensitive API library by verifying ~~the~~ a one of the at least one digital signature included in the software application that has a signature identification corresponding to the signature identifier of the sensitive API library.

72.   (Previously Presented) The code signing system of claim 57, wherein the digital signature is generated using a private signature key, and the virtual machine uses a public signature key to verify the authenticity of the digital signature.

73.   (Previously Presented) The code signing system of claim 72, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application; and

the virtual machine verifies the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

74. (Previously Presented) The code signing system of claim 60, wherein the API further comprises:

a description string that is displayed by the mobile device when the software application attempts to access the API.

75. (Previously Presented) The code signing system of claim 57, wherein the application platform comprises an operating system.

76 (Previously Presented) The code signing system of claim 57, wherein the application platform comprises one or more core functions of a mobile device.

77. (Previously Presented) The code signing system of claim 57, wherein the application platform comprises hardware on a mobile device.

78. (Previously Presented) The code signing system of claim 57, wherein the hardware comprises a subscriber identity module (SIM) card.

79. (Previously Presented) The code signing system of claim 57, wherein the software application is a Java application for a mobile device.

80. (Previously Presented) The code signing system of claim 57, wherein the API interfaces with a cryptographic routine on the application platform.

81. (Previously Presented) The code signing system of claim 57, wherein the API interfaces with a proprietary data model on the application platform.

82. (Currently Amended) The code signing system of claim 57, wherein the virtual machine is a ~~Java~~ virtual machine installed on a mobile device.

83. (Currently Amended) A method of controlling access to sensitive application programming interfaces on a mobile device, comprising the steps of:

loading a software application on the mobile device that requires access to a sensitive application programming interface (API) having a signature identifier;

determining whether the software application includes a digital signature and a signature identification; and

denying the software application access to the sensitive API ~~where~~ if the signature identification does not correspond with the signature identifier.

84. (Currently Amended) The method of claim 83, comprising the additional step of:

purging the software application from the mobile device ~~where~~ if the signature

identification does not correspond with the signature identifier.

85. (Previously Presented) The method of claim 83, wherein the digital signature and the

signature identification are generated by a code signing authority.

86. (Currently Amended) The method of claim 83, comprising the additional steps of:

verifying the authenticity of the digital signature ~~where~~ if the signature identification

corresponds with the signature identifier[[.]]; and

denying the software application access to the sensitive API ~~where~~ if the digital signature

is not authenticated.

87. (Currently Amended) The method of claim 86, comprising the additional step of:

purging the software application from the mobile device ~~where~~ if the digital signature is

not authenticated.

88. (Previously Presented) The method of claim 86, wherein the digital signature is

generated by applying a private signature key to a hash of the software application, and wherein

the step of verifying the authenticity of the digital signature is performed by a method

comprising the steps of:

storing a public signature key that corresponds to the private signature key on the mobile

device;

generating a hash of the software application to obtain a generated hash;

applying the public signature key to the digital signature to obtain a recovered hash; and

comparing the generated hash with the recovered hash.

89.   (Previously Presented)   The method of claim 88, wherein the digital signature is

generated by calculating a hash of the software application and applying the private signature

key.

90.   (Previously Presented)   The method of claim 83, comprising the additional step of:

displaying a description string that notifies a user of the mobile device that the software

application requires access to the sensitive API.

91.   (Previously Presented)   The method of claim 90, comprising the additional step of:

receiving a command from the user granting or denying the software application access

to the sensitive API.

92.   (Currently Amended) A method of controlling access to an application programming

interface (API) having a signature identifier on a mobile device by a software application created

by a software developer, comprising the steps of:

receiving the software application from the software developer;

determining whether the software application satisfies at least one criterion;

appending a digital signature and a signature identification to the software application

where if the software application satisfies at least one criterion;[[;]]

verifying the authenticity of the digital signature appended to the software application ~~where~~ if the signature identification corresponds with the signature identifier; and

providing access to the API to software applications ~~where~~ if the digital signature is authenticated.

93.   (Previously Presented)   The method of claim 92, wherein the step of determining whether the software application satisfies at least one criterion is performed by a code signing authority.

94.   (Previously Presented)   The method of claim 92, wherein the step of appending the digital signature and the signature identification to the software application includes generating the digital signature comprising the steps of:

calculating a hash of the software application; and

applying a signature key to the hash of the software application to generate the digital signature.

95.   (Previously Presented)   The method of claim 94, wherein the hash of the software application is calculated using the Secure Hash Algorithm (SHA1).

96.   (Previously Presented)   The method of claim 94, wherein the step of verifying the authenticity of the digital signature comprises the steps of:

providing a corresponding signature key on the mobile device;

calculating the hash of the software application on the mobile device to obtain a calculated hash;

applying the corresponding signature key to the digital signature to obtain a recovered

hash; and

authenticating the digital signature by comparing the calculated hash with the recovered

hash.

97.   (Currently Amended)   The method of claim 96, comprising the further step of denying

the software application access to the API ~~where~~ if the digital signature is not authenticated.

98.   (Previously Presented)   The method of claim 96, wherein the signature key is a private

signature key and the corresponding signature key is a public signature key.

99.   (Currently Amended)   A method of controlling access to a sensitive application

programming interface (API) having a signature identifier on a mobile device, comprising the

steps of:

registering one or more software developers that are trusted to develop software

applications which access the sensitive API;

receiving a hash of a software application;

determining whether the hash was sent by a registered software developer; and

generating a digital signature using the hash of the software application and a signature

identification corresponding to the signature identifier ~~where~~ if the hash was sent by the

registered software developer;

wherein

the digital signature and the signature identification are appended to the software

application; and

the mobile device verifies the authenticity of the digital signature in order to control

access to the sensitive API by the software application ~~where~~ if the signature identification

corresponds with the signature identifier.

100.   (Previously Presented)   The method of claim 99, wherein the step of generating the

digital signature is performed by a code signing authority.

101.   (Previously Presented)   The method of claim 99, wherein the step of generating the

digital signature is performed by applying a signature key to the hash of the software application.

102.   (Currently Amended)   The method of claim 101, wherein the mobile device verifies the

authenticity of the digital signature by performing the additional steps of:

providing a corresponding signature key on the mobile device;

calculating the hash of the software application on the mobile device to obtain a

calculated hash;

applying the corresponding signature key to the digital signature to obtain a recovered

hash;

determining whether the digital signature is authentic by comparing the calculated hash

with the recovered hash; and

denying the software application access to the sensitive API ~~where~~ if the digital signature

is not authenticated.

103. (Currently Amended) A method of restricting access to application programming interfaces on a mobile device, comprising the steps of:

loading a software application having a digital signature and a signature identification on the mobile device that requires access to one or more application programming interfaces (APIs) having at least one signature identifier;

authenticating the digital signature ~~where~~ if the signature identification corresponds with the signature identifier; and

denying the software application access to the one or more APIs ~~where~~ if the software application does not include an authentic digital signature[[ ]].

104. (Previously Presented) The method of claim 103, wherein the digital signature and signature identification are associated with a type of mobile device.

105. (Currently Amended) The method of claim 103, comprising the additional step of:

purging the software application from the mobile device ~~where~~ if the software application does not include an authentic digital signature. [[.]]

106. (Previously Presented) The method of claim 103, wherein:

the software application includes a plurality of digital signatures and signature identifications; and

the plurality of digital signatures and signature identifications includes digital signatures and signature identifications respectively associated with different types of mobile devices.

107. (Previously Presented)   The method of claim 106, wherein each of the plurality of digital signatures and associated signature identifications are generated by a respective corresponding code signing authority.

108. (Currently Amended)   The method of claim 103, wherein the step of determining whether the software application includes an authentic digital signature comprises the additional steps of:

verifying the authenticity of the digital signature where if the signature identification corresponds with respective ones of the at least one signature identifier.

109. (Previously Presented)   The method of claim 107, wherein each of the plurality of digital signatures and signature identifications are generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application.

110. (Currently Amended)   The method of claim 103, wherein the step of authenticating the digital signature where if the signature identification corresponds with the signature identifier comprises the steps of:

verifying that the signature identification corresponds with the signature identifier;

wherein authenticating the digital signature where if signature identification corresponds with the signature identifier comprising the steps of:

storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the digital signature;

generating a hash of the software application to obtain a generated hash;

applying the public signature key to the digital signature to obtain a recovered hash; and

comparing the generated hash with the recovered hash.

111.   (Currently Amended)   The method of claim 103, wherein:

the mobile device includes a plurality of APIs;

at least one of the plurality of APIs is classified as sensitive;

access to any of the plurality of APIs requires an authentic global signature;

access to each of the plurality of sensitive APIs requires an authentic global signature and an authentic digital signature associated with a signature identification;

the step of determining whether the software application includes an authentic digital signature and signature identification comprises the steps of:

determining whether the one or more APIs to which the software application requires access includes a sensitive API;

determining whether the software application includes an authentic global signature; and

determining whether the software application includes an authentic digital signature and signature identification ~~where~~ if the one or more APIs to which the software application requires access includes a sensitive API and the software application includes an authentic global signature; and

the step of denying the software application access to the one or more APIs comprises the steps of:

denying the software application access to the one or more APIs ~~where~~ if the software

application does not include an authentic global signature; and

denying the software application access to the sensitive API ~~where~~ if the one or more

APIs to which the software application requires access includes a sensitive API, the software

application includes an authentic global signature, and the software application does not include

an authentic digital signature and signature identifier required to access the sensitive API.

112.   (Currently Amended) A code signing system for controlling access to application

programming interfaces (APIs) having signature identificaters by software applications, the code

signing system comprising:

a verification system for executing on a processor and for authenticating digital

signatures ~~provided by~~ included in the respective software applications to access the APIs ~~where~~

if the signature ~~identifications~~ identifiers correspond with the signature ~~identificaters~~ identifier of

the respective APIs and ~~where~~ if a digital signature for a software application is generated with a

signature identification corresponding to a signature ~~identificater~~ identifier to access at least one

API; and

a control system for executing on a processor and for allowing access to at least one of

the APIs ~~where~~ if the digital signature ~~provided by~~ included in the software application is

authenticated by the verification system.

113.   (Previously Presented) The code signing system of claim 112, wherein a virtual machine

comprises the verification system and the control system.

114. (Previously Presented) The code signing system of claim 113, wherein the virtual machine is a Java virtual machine installed on a mobile device.

115. (Previously Presented) The code signing system of claim 112, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

116. (Previously Presented) The code signing system of claim 112, wherein the code signing system is installed on a mobile device and the software application is a Java application for a mobile device.

117. (Previously Presented) The code signing system of claim 112, wherein the digital signature and the signature identification of the software application are generated by a code signing authority.

118. (Previously Presented) The code signing system of claim 112, wherein the APIs access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

119. (Previously Presented) The code signing system of claim 112, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and the verification system uses a public signature key to authenticate the digital signature.

120.	(Previously Presented) The code signing system of claim 119, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

121.	(Previously Presented) The code signing system of claim 112, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

122.	(Previously Presented) The code signing system of claim 112, wherein the APIs provides access to at least one of one or more core functions of a mobile device, an operating system, and hardware on a mobile device.

123.	(Currently Amended) The code signing system of claim 112, wherein verification of a global digital signature provided by to the software application is required for accessing any of the APIs.

124. (Currently Amended) A method of controlling access to application programming interfaces (APIs) having signature identifiers by software applications, the method comprising:

executing instructions on a processor for authenticating digital signatures provided ~~by~~ to the respective software applications to access the APIs ~~where~~ if the signature identifications correspond with the signature identifiers of the respective APIs and ~~where~~ if a digital signature for a software application is generated with a signature identification corresponding to a signature identifier to access at least one API; and

executing instructions on a processor for allowing access to at least one of the APIs ~~where~~ if the digital signature provided ~~by~~ to the software application is authenticated.

125. (Currently Amended) The method of claim 124, wherein one digital signature and one signature identification are provided ~~by~~ to the software application access a library of at least one of the APIs.

126. (Previously Presented) The method of claim 124, wherein the digital signature and the signature identification of the software application are generated by a code signing authority.

127. (Previously Presented) The method of claim 124, wherein the APIs access at least one of a cryptographic module that implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

128.   (Previously Presented)   The method of claim 124, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and a public signature key is used to authenticate the digital signature.

129.   (Previously Presented)   The method of claim 128, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

the digital signature is authenticated by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

130.   (Previously Presented)   The method of claim 124, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

131.   (Previously Presented)   The method of claim 124, wherein the APIs provides access to at least one of one or more core functions of a mobile device, an operating system, and hardware on a mobile device.

132.   (Currently Amended)   The method of claim 124, wherein verification of a global digital signature provided by-to the software application is required for accessing any of the APIs.

133.   (Currently Amended)   A ~~management~~ system for controlling access by software

applications to application programming interfaces (APIs) having at least one signature identifier

on a subset of a plurality of mobile devices, the management system comprising:

a code signing authority for ~~providing~~ associating digital signatures and signature

identifications ~~to~~ with software applications that require access to at least one of the APIs with a

signature identifier on the subset of the plurality of mobile devices, ~~where~~ if a digital signature

for a software application is generated with a signature identification corresponding to a

signature identifier, and the signature identifications provided to the software applications

comprise those signature identifications that correspond to the signature identifiers that are

~~substantially only~~ on the subset of the plurality of mobile devices; wherein each mobile device of

the subset of the plurality of mobile devices comprises

a verification system for authenticating digital signatures provided ~~by~~ to the respective

software applications to access respective APIs ~~where~~ if the digital identifications correspond to

the digital identifiers of the respective APIs; and

a control system for allowing the respective software applications to access at least one of

the APIs ~~where~~ if the digital signatures provided ~~by~~ to the respective software applications are

authenticated by the verification system.

134.   (Currently Amended) The ~~management~~ system of claim 133, wherein a virtual machine

comprises the verification system and the control system.

135.   (Currently Amended) The ~~management~~ system of claim 134, wherein the virtual

machine is a Java virtual machine and the software applications are Java applications.

136. (Currently Amended) The ~~management~~ system of claim 133, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

137. (Currently Amended) The ~~management~~ system of claim 133, wherein the APIs access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

138. (Currently Amended) The ~~management~~ system of claim 133, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and the verification system uses a public signature key to authenticate the digital signature.

139. (Currently Amended) The ~~management~~ system of claim 138, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

140. (Currently Amended) The ~~management~~ system of claim 133, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

141. (Currently Amended) The ~~management~~ system of claim 133, wherein the subset of the plurality of mobile devices comprises mobile devices under the control of at least one of a corporation and a carrier.

142. (Currently Amended) The ~~management~~ system of claim 133, wherein a global digital signature provided ~~by~~ to the software application has to be authenticated before the software application is allowed access to any of the APIs on a mobile device of the subset of the plurality of mobile devices.

143. (Currently Amended) A method of controlling access by software applications to application programming interfaces (APIs) having at least one signature identifier on a subset of a plurality of mobile devices, the method comprising:

generating digital signatures for software applications with signature identifications corresponding to respective signature identifiers of the APIs; and

providing the digital signatures and the signature identifications to software applications that require access to at least one of the APIs on the subset of the plurality of mobile devices, ~~where~~ if the signature identifications provided to the software applications comprise those signature identifications that correspond to the signature identifiers that are ~~substantially only~~ on

the subset of the plurality of mobile devices; wherein each mobile device of the subset of the plurality of mobile devices comprises

a verification system for authenticating digital signatures provided ~~by~~ to the respective software applications to access respective APIs ~~where~~ if the digital identifications correspond to the digital identifiers of the respective APIs; and

a control system for allowing the software application to access at least one of the APIs ~~where~~ if the digital signature provided ~~by~~ to the software application is authenticated by the verification system.

144. (Previously Presented)   The method of claim 143, wherein a virtual machine comprises the verification system and the control system.

145. (Previously Presented)   The method of claim 144, wherein the virtual machine is a Java virtual machine and the software applications are Java applications.

146. (Previously Presented)   The method of claim 143, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

147. (Previously Presented)   The method of claim 143, wherein the APIs access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

148. (Previously Presented)   The method of claim 143, wherein at least one of the digital signatures is generated using a private signature key under a signature scheme associated with a signature identification, and the verification system uses a public signature keys to authenticate said at least one of the digital signatures.

149. (Previously Presented)   The method of claim 148, wherein:

at least one of the digital signatures is generated by applying the private signature key to a hash of a software application under the signature scheme; and

the verification system authenticates said at least one of the digital signatures by generating a hash of the software application to obtain a generated hash, applying the public signature key to said at least one of the digital signatures to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

150. (Previously Presented)   The method of claim 143, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

151. (Previously Presented)   The method of claim 143, wherein the subset of the plurality of mobile devices comprises mobile devices under the control of at least one of a corporation and a carrier.

152.   (Currently Amended) A mobile device ~~for a subset of a plurality of mobile devices, the mobile device~~ comprising:

an application platform having application programming interfaces (APIs);

a verification system for authenticating digital signatures and signature identifications provided ~~by~~ to the respective software applications to access the APIs; and

a control system for allowing a software application to access at least one of the APIs ~~where~~ if a digital signature provided ~~by~~ to the software application is authenticated by the verification system;

wherein a code signing authority provides digital signatures and signature identifications to software applications that require access to at least one of the APIs such that the digital signature for the software application is generated according to a signature scheme of a signature identification, and wherein the signature identifications provided to the software applications comprise those signature identifications that are ~~substantially only~~ authorized to allow access on ~~the~~ a subset of ~~the~~ a plurality of mobile devices.

153.   (Previously Presented) The mobile device of claim 152, wherein a virtual machine comprises the verification system and the control system.

154.   (Previously Presented) The mobile device of claim 153, wherein the virtual machine is a Java virtual machine and the software application is a Java application.

155.   (Previously Presented) The mobile device of claim 152, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

156.   (Previously Presented) The mobile device of claim 152, wherein the APIs of the application platform access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

157.   (Previously Presented) The mobile device of claim 152, wherein the digital signature is generated using a private signature key under the signature scheme, and the verification system uses a public signature key to authenticate the digital signature.

158.   (Previously Presented) The mobile device of claim 157, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

159.   (Previously Presented) The mobile device of claim 152, wherein at least one of the APIs further comprises:

Page 917 of 1415

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

160. (Currently Amended) A method of controlling access to application programming interfaces (APIs) of an application platform of a mobile device ~~for a subset of a plurality of mobile devices~~, the method comprising:

receiving digital signatures and signature identifications from software applications that require ~~to~~ access to the APIs~~:~~

authenticating the digital signatures and the signature identifications; and

allowing a software application to access at least one of the APIs ~~where~~ if a digital signature provided ~~by~~ to the software application is authenticated;

wherein a code signing authority provides the digital signatures and the signature identifications to the software applications that require access to at least one of the APIs such that the digital signature for the software application is generated according to a signature scheme of a signature identification, and wherein the signature identifications provided to the software applications comprise those signature identifications that are ~~substantially only~~ authorized to allow access on ~~the~~ a subset of ~~the~~ a plurality of mobile devices.

161. (Previously Presented) The method of claim 160, wherein one digital signature and one signature identification is required for accessing each library of at least one of the APIs.

Page 918 of 1415

162. (Previously Presented) The method of claim 160, wherein the APIs of the application platform access at least one of a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI).

163. (Previously Presented) The method of claim 160, wherein the digital signature is generated using a private signature key under the signature scheme, and a public signature key is used to authenticate the digital signature.

164. (Previously Presented) The method of claim 163, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

the digital signature is authenticated by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same.

165. (Previously Presented) The method of claim 160, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs.

REMARKS

Claims 57-165 are pending in this application. Claims 57, 62, 83, 92, 99, 103, 112, 124, 133, 143, 152, and 160 are independent claims. The pending claims stand rejected by the examiner. Assignee traverses the instant rejections.

Information Disclosure Statement

The office action provided that the non-patent literature document: "Handbuch der Chinkarten" was not considered due to a lack of an English translation being provided. Information regarding this document will be supplied under separate cover.

Claim Objections

Claims 86, 92, 105, 112, and 132 were objected to because of certain informalities. The assignee has amended these claims per the recommendations in the office action. Accordingly, the assignee respectfully submits that the instant claim objections be withdrawn.

Claim Rejections-35 U.S.C. § 112

Claims 82, 133, 143, 152, and 160 stand rejected under 35 U.S.C. § 112 because of use of certain terms in these claims: the trademark/trade name "Java," "substantially," and others. The assignee has amended these claims to address the use of these terms. Accordingly, the assignee respectfully submits that the instant claim rejections be withdrawn.

Claim Rejections-35 U.S.C. § 101

Claims 112-132 stand rejected under 35 U.S.C. § 101 because the office action maintains that the claimed invention is directed to non-statutory subject matter. The assignee respectfully disagrees, but in order to expedite prosecution of this application, amendments have made to these claims to emphasize the processor-based environment of the claimed subject matter. Accordingly, the assignee respectfully submits that the instant claim rejections be withdrawn.

Claim Rejections-35 U.S.C. §§ 102 and 103

The independent claims 57, 62, 83, 92, 99, 103, 112, 124, 133, 143, 152, and 160 stand rejected under 35 U.S.C. § 102(e) as being anticipated by United States Patent No. 6,795,9192 to Gibbs et al., hereinafter Gibbs. The dependent claims are rejected based upon Gibbs alone under Section 102 or in combination with other cited references under Section 103. These rejections are traversed.

Claim 57 is directed to a code signing system for use with a software application having a digital signature and a signature identification. An application programming interface (API) has an associated signature identifier and is configured to link the software application with the application platform. A virtual machine verifies the authenticity of the digital signature in order to control access to the API by the software application if the signature identifier corresponds to the signature identification.

As shown by the foregoing, claim 57 requires *an application programming interface (API) to have an associated signature identifier*. The office action maintains that this subject matter is disclosed by Gibbs at column 5, lines 40-51, which read as follows:

> However, authenticated message server 316 is alternatively part of the
> software component stack added to server 308. In such an embodiment, an

> application programming interface ("API") for the messaging server 308
> is added which provides access to the authenticated message server
> services. Authenticated message server services include generating and
> authenticating unique digital signatures as described herein. The unique
> digital signature system 300 can be highly distributed, wherein incoming
> and outgoing messages are handled by separate servers or computer
> systems on an interconnected 50 network (e.g. a LAN).

This passage of Gibbs merely discloses that an API can be used to access services that include "generating and authenticating unique digital signatures." There is no disclosure that an API has an associated signature identifier as required in claim 57. The API in this passage from Gibbs only allows a digital signature to be generated, not that there is a pre-existing signature identifier which is already associated with an API. Because of such lack of disclosure, claim 57 is allowable and should proceed to issuance.

Still further, claim 57 requires that *a virtual machine verifies the authenticity of the digital signature in order to control access to the API by the software application if the signature identifier corresponds to the signature identification.* The office action maintains that such subject matter of claim 57 is disclosed in the following passages of Gibbs:

> Functionally, WWW server 424 hosts a website with an electronic
> commerce application and, preferably, an interface (e.g., Perl, CGI,
> HTML, Java, ASP, ODBC, etc.) to authenticated message server 428.
> According to one embodiment, WWW server 424 is preferably a Sun
> Microsystems SPARC™ system, running WWW/Internet server software
> from Netscape Corporation. A remote user, for example a user on laptop
> 452, which is connected to the Internet 444 via an internet access provider
> ("TAP") or local area network ("LAN") 448, is typically connect to
> WWW server 424 through a dedicated communications port over the
> Internet 444. Once connected, the remote user at laptop 452 can either
> purchase a unique digital signature 132, or request a particular piece of
> media or service from the unique digital signature system 400 using a
> unique digital signature 132 and the interface on WWW server 424.
> [(See, Gibbs at col. 6, lines 45-60.)]
>
> Outbound media includes, but is not limited to: ASCII text, HTML files,
> Java applets, WAV files, AVI files, MPEG files and the like. In one

embodiment, message server 408 is a wireless short message/paging
service ("SMS"), ...
[(See, Gibbs at col. 7, lines 2-5.)]

If the unique digital signature 132 cannot be validated, that is it is rejected
by the authentication process "A", then an error message is returned to the
requestor. If the unique digital signature 132 is validated by the
authentication process "A", that is it is accepted, then processing continues
to step 712. In step 712, the successfully validated unique digital signature
132 is forwarded to a particular username or automated process
(servicename). Next, in step 716, the request identified by the unique
digital signature 132 is processed by a stored procedure, or by a local user,
as the case may be. Finally, at step 720, a response message is returned to
the remote user of the unique digital signature 132. According to one
embodiment, media files are returned, such as Java applets, one or more
bundled HTML files, an MPEG file, a WAV file, or a RAM file. In
another embodiment, the media files are stored on a proxy server 420 and
accessed at the proxy server 420 by the remote user.
[...]
In one embodiment, the unique digital signature 132 is sent to a remote
user (e.g., 452) via WWW server 424. In another embodiment the unique
digital signature 132 is sent via electronic message server 408. In still
another embodiment, the unique digital signature 132 is sent via "snail
mail" to the remote user's personal home address.
[(See, Gibbs at col. 10, lines 14-30 and 35-45.)]

These passages of Gibbs are part of a manual user process wherein "a remote user in possession

of a unique digital signature 132 may wish to purchase something with the unique digital

signature 132" or participate "in an electronic voting or polling system." (See, Gibbs

respectively at: col. 9, line 67 to col. 10, line 2; and col. 10, lines 32-33.) These passages of

Gibbs disclose what occurs "[i]f the unique digital signature 132 cannot be validated" or "if the

unique digital signature 132 is validated." For example in the user purchase illustration of

Gibbs, if the unique digital signature can be validated, then "media files are returned." However,

there is no disclosure in these passages of any authentication processing being based upon

whether *the signature identifier corresponds to the signature identification* of claim 57, let alone

a disclosure of any authentication processing being based upon whether *the signature identifier*

(of an API) corresponds to the signature identification *(of a software application)* as required by claim 57. Because of such lack of disclosure, claim 57 is allowable and should proceed to issuance.

With respect to the other independent claims, these claims recite similar subject matter as claim 57. Accordingly, these claims are allowable for the reasons provided above with respect to claim 57.

Still further, the assignee disagrees with other positions in the office action. For example, the assignee disagrees with the rejection of claim 59. Claim 59 depends from independent claim 57 and recites that the *virtual machine purges the software application if the digital signature is not authenticated.* The office action maintains that this subject matter is disclosed in Gibbs at col. 10, lines 63-67, which read as follows:

> If the unique digital signature 132 was discarded at step 708, one of at least three responses (step 724) is appropriate. One response is to completely disregard the failed unique digital signature 132 and voting responses and delete them from the unique digital signature system 400.
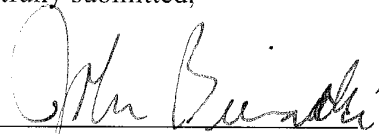
This passage from Gibbs discloses discarding a <u>digital signature</u> and <u>voting responses</u> from a user. There is no disclosure that there is a discarding of a *software application.* Because of such lack of disclosure, claim 59 is allowable and should proceed to issuance.

Assignee at this time has not provided arguments in support of the patentability of other dependent claims. It is respectfully submitted that because the independent claims are now in condition for allowance, these dependent claims which depend directly or indirectly therefrom are also in condition for allowance. However, assignee reserves the right to argue the patentability of certain of the dependent claims in the instant application at a future time, should that become necessary.

CONCLUSION

For the foregoing reasons, assignee respectfully submits that the pending claims are allowable. Therefore, the examiner is respectfully requested to pass this case to issue. If the examiner believes it would be useful to advance prosecution, the examiner is invited to telephone the undersigned at (216) 586-3939.

Respectfully submitted,

John V. Biernacki
Reg. No. 40,511
Jones, Day
North Point
901 Lakeside Avenue
Cleveland, OH 44114-1190
(216) 586-7747

Electronic Acknowledgement Receipt

EFS ID:	5513466
Application Number:	10381219
International Application Number:	
Confirmation Number:	9761
Title of Invention:	Software code signing system and method
First Named Inventor/Applicant Name:	David P Yach
Correspondence Address:	David B Cochran Jones Day North Point 901 Lakeside Avenue Cleveland OH 44114-1190 US - -
Filer:	Stephen D. Scanlon/John V. Biernacki
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	555255012423
Receipt Date:	15-JUN-2009
Filing Date:	20-MAR-2003
Time Stamp:	11:48:55
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Amendment/Req. Reconsideration-After Non-Final Reject	DOC006.pdf	1145284 bc91f3b2feeb10adab46495eafbbe6fd6cd9a164	no	35

Warnings:

Information:

	Total Files Size (in bytes):	1145284

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
10/381,219	03/20/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐	OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))						
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	

APPLICATION AS AMENDED – PART II

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY	OR	OTHER THAN SMALL ENTITY	
06/15/2009	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 109	Minus	** 109	= 0	X $ =		OR X $52=	0
Independent (37 CFR 1.16(h))	* 12	Minus	*** 12	= 0	X $ =		OR X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))								
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
					TOTAL ADD'L FEE		OR TOTAL ADD'L FEE	0

	(Column 1)		(Column 2)	(Column 3)				
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR X $ =	
☐ Application Size Fee (37 CFR 1.16(s))								
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR		
					TOTAL ADD'L FEE		OR TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/LINDA HUMES/

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	555255012423	9761

7590   03/17/2009

David B Cochran
Jones Day
North Point
901 Lakeside Avenue
Cleveland, OH 44114-1190

EXAMINER
AVERY, JEREMIAH L

ART UNIT	PAPER NUMBER
2431	

MAIL DATE	DELIVERY MODE
03/17/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

<table>
<tr>
<td rowspan="2">Office Action Summary</td>
<td>Application No.
10/381,219</td>
<td colspan="2">Applicant(s)
YACH ET AL.</td>
</tr>
<tr>
<td>Examiner
JEREMIAH AVERY</td>
<td>Art Unit
2431</td>
<td></td>
</tr>
</table>

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>25 May 2007</u>.
2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>57-165</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) <u>57-165</u> is/are rejected.
7)☒ Claim(s) <u>86,92,105,112 and 132</u> is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on <u>20 March 2003</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☒ All b)☐ Some * c)☐ None of:
        1.☐ Certified copies of the priority documents have been received.
        2.☐ Certified copies of the priority documents have been received in Application No. _____.
        3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-SB/08)
    Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   Claims 1-56 have been cancelled in a preliminary amendment.

2.   Claims 57-165 have been added in a preliminary amendment.

3.   Claims 57-165 have been examined.

Information Disclosure Statement

1.   The NPL document: "Handbuch der Chinkarten" was not considered due to a lack of an

English translation being provided.

Claim Objections

2.   Claims 86, 92, 105 and 132 are objected to because of the following informalities:

punctuation errors. Claim 86 has a period (.) after the claim language "the signature identifier"

which also has a semi-colon after said claim language. Claim 92 has an extra semi-colon after

the claim language "at least one criterion". Claim 105 has an extra period at the conclusion of

the claim language. Claim 132 is missing a period at the conclusion of the claim language.

Appropriate correction is required.

3.   Claim 112 is objected to because of the following informalities: spelling error. Claim 112

has the term "identificaters". The Examiner will broadly interpret this term to be "identifiers".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and
distinctly claiming the subject matter which the applicant regards as his invention.

4.   Claims 82, 133, 143, 152 and 160 are rejected under 35 U.S.C. 112, second paragraph,

as being indefinite for failing to particularly point out and distinctly claim the subject matter which

applicant regards as the invention.

5.   Claim 82 contains the trademark/trade name "Java". Where a trademark or trade name

is used in a claim as a limitation to identify or describe a particular material or product, the claim

does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte*

Simpson, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark

or trade name cannot be used properly to identify any particular material or product. A

trademark or trade name is used to identify a source of goods, and not the goods themselves.

Thus, a trademark or trade name does not identify or describe the goods associated with the

trademark or trade name.

6.   Within claims 133 and 143, the term "signature identifiers that are *substantially* only on

the subset of the plurality of mobile devices" is a relative term which renders the claim indefinite.

The term "signature identifiers that are *substantially* only on the subset of the plurality of mobile

devices" is not defined by the claim, the specification does not provide a standard for

ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably

apprised of the scope of the invention.

7.   Within claims 152 and 160, the term "signature identifications that *are substantially* only

authorized to allow access" is a relative term which renders the claim indefinite. The term

"signature identifications that are *substantially* only authorized to allow access" is not defined by

the claim, the specification does not provide a standard for ascertaining the requisite degree,

and one of ordinary skill in the art would not be reasonably apprised of the scope of the

invention.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or
composition of matter, or any new and useful improvement thereof, may obtain a patent
therefor, subject to the conditions and requirements of this title.

8.   Claims 112-132 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Independent claim 112 possesses "a code signing system" but said "system" lacks a tangible embodiment within the claim language. Further, independent claim 124 possesses "a method of controlling access to application programming interfaces (APIs) having signature identifiers by software applications", however said "application programming interfaces (APIs)" are not specified as to whether they are a tangible embodiment. Though within the Specification said API interacts with hardware, it is not distinct as to whether they are hardware themselves or just software applications. Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 57-71, 74-77, 79-87, 90-97, 99-108, 11-118, 121-127, 130-137, 140-147, 150-156, 159-162 and 165 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,795,919 to Gibbs et al., hereinafter Gibbs.

9.   (New) Regarding claim 57, Gibbs discloses a code signing system for operation in conjunction with a software application having a digital signature and a signature identification, where the digital signature is associated with the signature identification, comprising:

an application platform (column 3, lines 10-18, "an authenticated message server functionally

comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server

316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-

66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62);

an application programming interface (API) having an associated signature identifier, the API is

configured to link the software application with the application platform (column 5, lines 40-51,

"an application server interface ('API') for messaging server 308 is added which provides

access to the authenticated message server services");

and a virtual machine that verifies the authenticity of the digital signature in order to control

access to the API by the software application where the signature identifier corresponds to the

signature identification (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10,

lines 14-30 and 35-45).

10.   (New) Regarding claim 58, Gibbs discloses wherein the virtual machine denies the

software application access to the API if the digital signature is not authenticated (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

11.   (New) Regarding claim 59, Gibbs discloses, wherein the virtual machine purges the

software application if the digital signature is not authenticated (column 10, lines 63-67, "One

response is to completely disregard the failed unique digital signature 132 and voting responses

and delete them from the unique digital signature system 400").

12.   (New) Regarding claim 60, Gibbs discloses, wherein the code signing system is

installed on a mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column

10, lines 8-13).

13.   (New) Regarding claim 61, Gibbs discloses, wherein the digital signature is generated by a code signing authority (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

14.   (New) Regarding claim 62, Gibbs discloses a code signing system for operation in conjunction with a software application having a digital signature and a signature identification where the digital signature is associated with the signature identification, comprising:

an application platform (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62);

a plurality of application programming interfaces (APIs) associated with a signature identifier, each configured to link the software application with a resource on the application platform (column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is added which provides access to the authenticated message server services");

a virtual machine that verifies the authenticity of the digital signature in order to control access to the APIs by the software application where the signature identification corresponds to the signature identifier, wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

15.   (New) Regarding claim 63, Gibbs discloses, wherein the plurality of APIs are included in an API library (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 26-30 and column 11, lines 8-12, "add-on software component in servers 424 or 408").

16.   (New) Regarding claim 64, Gibbs discloses, wherein one or more of the plurality of APIs

is classified as sensitive and having an associated signature identifier, and wherein the virtual

machine uses the digital signature and the signature identification to control access to the

sensitive APIs (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67,

"electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7).

17.   (New) Regarding claim 65, Gibbs discloses, wherein the code signing system operates

in conjunction with a plurality of software applications, wherein one or more of the plurality of

software applications has a digital signature and a signature identification, and wherein the

virtual machine verifies the authenticity of the digital signature of each of the one or more of the

plurality of software applications, where the signature identification corresponds to the signature

identifier of the respective sensitive APIs, in order to control access to the sensitive APIs by

each of the plurality of software applications (column 8, lines 56-67, column 9, lines 1-5 and 45-

50, column 10, lines 31-67, "electronic voting or polling system. In such a system, the unique

digital signature 132 is allocated by the authenticated message server 428 at step 704" and

column 11, lines 1-7).

18.   (New) Regarding claim 66, Gibbs discloses, wherein the resource on the application

platform comprises a wireless communication system (Figure 3, element 332 and Figure 4,

element 452, column 5, lines 29-33, "a wireless interconnection" and column 7, lines 4-8).

19.   (New) Regarding claim 67, Gibbs discloses, wherein the resource on the application

platform comprises a cryptographic module which implements cryptographic algorithms (column

8, lines 11-32 and column 9, lines 36-50).

20.   (New) Regarding claim 68, Gibbs discloses, wherein the resource on the application

platform comprises a data store (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-

62).

21.   (New) Regarding claim 69, Gibbs discloses, wherein the resource on the application

platform comprises a user interface (UI) (column 3, lines 10-18, "an authenticated message

server functionally comprises a digital service engine 120", column 5, lines 29-51,

"Authenticated message server 316 can run on a standard personal computer" and lines 60-65,

column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column

10, lines 49-62).

[A "personal computer" contains the means for a user interface.]

22.   (New) Regarding claim 70, Gibbs discloses further comprising:

a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs,

wherein the virtual machine controls access to the plurality of API libraries by the software

application (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and

40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and

column 11, lines 8-12, "add-on software component in servers 424 or 408").

23.   (New) Regarding claim 71, Gibbs discloses, wherein at least one of the plurality of API

libraries is classified as sensitive (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column

10, lines 31-67, "electronic voting or polling system. In such a system, the unique digital

signature 132 is allocated by the authenticated message server 428 at step 704" and column

11, lines 1-7);

wherein access to a sensitive API library requires a digital signature associated with a signature

identification where the signature identification corresponds to a signature identifier associated

with the sensitive API library (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10,

lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature

132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-

7);

wherein the software application includes at least one digital signature and at least one

associated signature identification for accessing sensitive API libraries (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7);

wherein the virtual machine authenticates the software application for accessing the sensitive

API library by verifying the one digital signature included in the software application that has a

signature identification corresponding to the signature identifier of the sensitive API library

(column 10, lines 14-30 and 35-45).

24.   (New) Regarding claim 74, Gibbs discloses, wherein the API further comprises:

a description string that is displayed by the mobile device when the software application

attempts to access the API (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

25.   (New) Regarding claim 75, Gibbs discloses, wherein the application platform comprises

an operating system (column 3, lines 10-18, "an authenticated message server functionally

comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server

316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer" and column 10, lines 49-62).

26.   (New) Regarding claim 76, Gibbs discloses, wherein the application platform comprises one or more core functions of a mobile device (column 6, lines 45-60, column 8, lines 48-55 and column 10, lines 8-13).

27.   (New) Regarding claim 77, Gibbs discloses, wherein the application platform comprises hardware on a mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

28.   (New) Regarding claim 79, Gibbs discloses, wherein the software application is a Java application for a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

29.   (New) Regarding claim 80, Gibbs discloses, wherein the API interfaces with a cryptographic routine on the application platform (column 8, lines 11-32 and column 9, lines 36-50).

30.   (New) Regarding claim 81, Gibbs discloses, wherein the API interfaces with a proprietary data model on the application platform (column 10, lines 31-67, "electronic voting or polling system" and column 11, lines 1-12).

31.   (New) Regarding claim 82, Gibbs discloses, wherein the virtual machine is a Java virtual machine installed on a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

32.   (New) Regarding claim 83, Gibbs teaches a method of controlling access to sensitive application programming interfaces on a mobile device, comprising the steps of:

loading a software application on the mobile device that requires access to a sensitive application programming interface (API) having a signature identifier (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts the remote user for access to

any unique digital signatures stored in a cookie file on laptop 452" and lines 31-67, "electronic

voting or polling system. In such a system, the unique digital signature 132 is allocated by the

authenticated message server 428 at step 704" and column 11, lines 1-7);

determining whether the software application includes a digital signature and a signature

identification (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-67,

"electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7);

denying the software application access to the sensitive API where the signature identification

does not correspond with the signature identifier (Figure 6, column 8, lines 56-65 and column

10, lines 14-30).

33.   (New) Regarding claim 84, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device where the signature

identification does not correspond with the signature identifier (column 10, lines 63-67, "One

response is to completely disregard the failed unique digital signature 132 and voting responses

and delete them from the unique digital signature system 400").

34.   (New) Regarding claim 85, Gibbs teaches, wherein the digital signature and the

signature identification are generated by a code signing authority (column 3, lines 8-18, 25-40

and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

35.   (New) Regarding claim 86, Gibbs teaches, comprising the additional steps of:

verifying the authenticity of the digital signature where the signature identification

corresponds with the signature identifier (Figure 6, column 8, lines 56-65 and column 10, lines

14-30);

denying the software application access to the sensitive API where the digital signature is not

authenticated (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

36.   (New) Regarding claim 87, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device where the digital signature is not

authenticated (column 10, lines 63-67, "One response is to completely disregard the failed

unique digital signature 132 and voting responses and delete them from the unique digital

signature system 400").

37.   (New) Regarding claim 90, Gibbs teaches, comprising the additional step of:

displaying a description string that notifies a user of the mobile device that the software

application requires access to the sensitive API (column 8, lines 28-39, "a limited character

ASCII set is used since remote users on legacy electronic message and existing telephone

systems can still type the unique digital signature without special software (or hardware)" and

column 10, lines 14-30, "media files are returned, such as Java applets, one or more bundled

HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

38.   (New) Regarding claim 91, Gibbs teaches, comprising the additional step of:

receiving a command from the user granting *or* denying the software application access to the

sensitive API (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

39.   (New) Regarding claim 92, Gibbs teaches a method of controlling access to an

application programming interface (API) having a signature identifier on a mobile device by a

software application created by a software developer, comprising the steps of:

receiving the software application from the software developer (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45);

determining whether the software application satisfies at least one criterion (column 7, lines 23-

38 and column 9, lines 51-58, "other processing/handling of the request for service continues -

depending on the particular application where the unique digital signature is employed");

appending a digital signature and a signature identification to the software application where the

software application satisfies at least one criterion (column 8, lines 11-27, "the digital signature

123 generated at step 516 is converted from a binary value to a value acceptable for electronic

messaging, i.e., ASCII text, by adaptation algorithm 128", column 8, lines 56-65, column 10,

lines 14-30 and column 11, lines 41-43);

verifying the authenticity of the digital signature appended to the software application where the

signature identification corresponds with the signature identifier (Figure 6, column 8, lines 56-65

and column 10, lines 14-30);

providing access to the API to software applications where the digital signature is authenticated

(column 5, lines 40-51, "an application server interface ('API') for messaging server 308 is

added which provides access to the authenticated message server services" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62).

40.   (New) Regarding claim 93, Gibbs teaches, wherein the step of determining whether the

software application satisfies at least one criterion is performed by a code signing authority

(Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

41.   (New) Regarding claim 94, Gibbs teaches, wherein the step of appending the digital

signature and the signature identification to the software application includes generating the

digital signature comprising the steps of:

calculating a hash of the software application (column 3, lines 49-58, "the input to the one-way

hash function" and column 8, lines 11-34, "MD5 function" and "SHA-1 hash function");

applying a signature key to the hash of the software application to generate the digital signature

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50).

42.   (New) Regarding claim 95, Gibbs teaches, wherein the hash of the software application

is calculated using the Secure Hash Algorithm (SHA-1) (column 8, lines 11-27).

43.   (New) Regarding claim 96, Gibbs teaches, wherein the step of verifying the authenticity

of the digital signature comprises the steps of:

providing a corresponding signature key on the mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13);

calculating the hash of the software application on the mobile device to obtain a calculated hash

(column 3, lines 49-58, "the input to the one-way hash function" and column 8, lines 11-34,

"MD5 function" and "SHA-1 hash function");

applying the corresponding signature key to the digital signature to obtain a recovered hash

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50);

authenticating the digital signature by comparing the calculated hash with the recovered hash

(column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital signature 144, is

compared against the adapted digital signature in the incoming unique digital signature").

44.   (New) Regarding claim 97, Gibbs teaches, comprising the further step of denying the

software application access to the API where the digital signature is not authenticated (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

45.   (New) Regarding claim 99, Gibbs teaches a method of controlling access to a sensitive

application programming interface (API) having a signature identifier on a mobile device,

comprising the steps of:

registering one or more software developers that are trusted to develop software applications

which access the sensitive API (column 2, lines 28-35, "a value derived from the hash is

concatenated with a service id. In one embodiment, the service id is a local username", column

3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and 40-47 and column 9, lines 41-50);

receiving a hash of a software application (column 2, lines 28-35, column 3, lines 49-67, column

4, lines 1-4, column 8, lines 19-32 and 40-47 and 56-65, column 9, lines 41-50 and column 10,

lines 14-30);

determining whether the hash was sent by a registered software developer (Figure 6, column 8,

lines 56-65, "the service id 104 is tested to verify that it represents a valid local username or

service name" and column 10, lines 14-30);

generating a digital signature using the hash of the software application and a signature

identification corresponding to the signature identifier where the hash was sent by the registered

software developer (column 3, lines 49-58, "the input to the one-way hash function" and column

8, lines 11-34, "MD5 function" and "SHA-1 hash function");

wherein the digital signature and the signature identification are appended to the software

application (column 8, lines 11-27, "the digital signature 123 generated at step 516 is converted

from a binary value to a value acceptable for electronic messaging, i.e., ASCII text, by

adaptation algorithm 128", column 8, lines 56-65, column 10, lines 14-30, "media files are

returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a

RAM file" and lines 49-62 and column 11, lines 41-43);

the mobile device verifies the authenticity of the digital signature in order to control access to the

sensitive API by the software application where the signature identification corresponds with the

signature identifier (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 31-

67, "electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7).

46.   (New) Regarding claim 100, Gibbs teaches, wherein the step of generating the digital

signature is performed by a code signing authority (column 3, lines 8-18, 25-40 and 64-67,

column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

47.   (New) Regarding claim 101, Gibbs teaches, wherein the step of generating the digital

signature is performed by applying a signature key to the hash of the software application

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50).

48.   (New) Regarding claim 102, Gibbs teaches, wherein the mobile device verifies the

authenticity of the digital signature by performing the additional steps of:

providing a corresponding signature key on the mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13);

calculating the hash of the software application on the mobile device to obtain a calculated hash

(column 3, lines 49-58, "the input to the one-way hash function" and column 8, lines 11-34,

"MD5 function" and "SHA-1 hash function");

applying the corresponding signature key to the digital signature to obtain a recovered hash

(column 2, lines 28-35, column 3, lines 49-67, column 4, lines 1-4, column 8, lines 19-32 and

40-47 and column 9, lines 41-50);

determining whether the digital signature is authentic by comparing the calculated hash with the

recovered hash (column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital signature

144, is compared against the adapted digital signature in the incoming unique digital signature");

denying the software application access to the sensitive API where the digital signature is not

authenticated (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

49.   (New) Regarding claim 103, Gibbs teaches a method of restricting access to application

programming interfaces on a mobile device, comprising the steps of:

loading a software application having a digital signature and a signature identification on the

mobile device that requires access to one or more application programming interfaces (APIs)

having at least one signature identifier (column 8, lines 11-27, "the digital signature 123

generated at step 516 is converted from a binary value to a value acceptable for electronic

messaging, i.e., ASCII text, by adaptation algorithm 128", column 8, lines 56-65, column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62 and column 11, lines 41-43);

authenticating the digital signature where the signature identification corresponds with the

signature identifier (column 8, lines 56-65 and column 9, lines 36-58, "an adapted digital

signature 144, is compared against the adapted digital signature in the incoming unique digital

signature");

denying the software application access to the one or more APIs where the software application

does not include an authentic digital signature (Figure 6, column 8, lines 56-65 and column 10,

lines 14-30).

50.　　(New) Regarding claim 104, Gibbs teaches, wherein the digital signature and signature

identification are associated with a type of mobile device (column 6, lines 45-60, "laptop",

column 8, lines 48-55 and column 10, lines 8-13).

51.　　(New) Regarding claim 105, Gibbs teaches, comprising the additional step of:

purging the software application from the mobile device where the software application does not

include an authentic digital signature (column 10, lines 63-67, "One response is to completely

disregard the failed unique digital signature 132 and voting responses and delete them from the

unique digital signature system 400").

52.　　(New) Regarding claim 106, Gibbs teaches, wherein:

the software application includes a plurality of digital signatures and signature

identifications (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19

and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and

column 11, lines 8-12, "add-on software component in servers 424 or 408");

the plurality of digital signatures and signature identifications includes digital signatures and

signature identifications respectively associated with different types of mobile devices (column

4, lines 19-42, "log file", "each of the unique digital signatures" and "100 records" and 50-65,

column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines

14-30 and 35-49).

53.   (New) Regarding claim 107, Gibbs teaches, wherein each of the plurality of digital

signatures and associated signature identifications are generated by a respective corresponding

code signing authority (column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5,

lines 45-51 and column 8, lines 40-47).

54.   (New) Regarding claim 108, Gibbs teaches, wherein the step of determining whether

the software application includes an authentic digital signature comprises the additional steps

of:

verifying the authenticity of the digital signature where the signature identification corresponds

with respective ones of the at least one signature identifier (Figure 6, column 8, lines 56-65 and

column 10, lines 14-30).

55.   (New) Regarding claim 111, Gibbs teaches, wherein:

the mobile device includes a plurality of APIs;

at least one of the plurality of APIs is classified as sensitive (column 8, lines 56-67, column 9,

lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In such a

system, the unique digital signature 132 is allocated by the authenticated message server 428

at step 704" and column 11, lines 1-7);

access to any of the plurality of APIs requires an authentic global signature (column 5, lines 40-

51, "an application server interface ('API') for messaging server 308 is added which provides

access to the authenticated message server services" and column 10, lines 14-30, "media files

are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file,

or a RAM file" and lines 49-62);

access to each of the plurality of sensitive APIs requires an authentic global signature and an

authentic digital signature associated with a signature identification (column 8, lines 56-67,

column 9, lines 1-5 and 45-50, column 10, lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7);

    the step of determining whether the software application includes an authentic digital

signature and signature identification comprises the steps of:

determining whether the one or more APIs to which the software application requires access

includes a sensitive API (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines

31-67, "electronic voting or polling system. In such a system, the unique digital signature 132 is

allocated by the authenticated message server 428 at step 704" and column 11, lines 1-7);

determining whether the software application includes an authentic global signature (column 5,

lines 40-51, "an application server interface ('API') for messaging server 308 is added which

provides access to the authenticated message server services" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62);

determining whether the software application includes an authentic digital signature and

signature identification where the one or more APIs to which the software application requires

access includes a sensitive API and the software application includes an authentic global

signature (column 5, lines 40-51, "an application server interface ('API') for messaging server

308 is added which provides access to the authenticated message server services" and column

10, lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML

files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

the step of denying the software application access to the one or more APIs comprises the

steps of:

denying the software application access to the one or more APIs where the software application

does not include an authentic global signature (Figure 6, column 8, lines 56-65 and column 10,

lines 14-30);

denying the software application access to the sensitive API where the one or more APIs to

which the software application requires access includes a sensitive API, the software application

includes an authentic global signature, and the software application does not include an

authentic digital signature and signature identifier required to access the sensitive API (Figure 6,

column 8, lines 56-65 and column 10, lines 14-30).

56.   (New) Regarding claim 112, Gibbs discloses a code signing system for controlling

access to application programming interfaces (APIs) having signature identificaters by software

applications, the code signing system comprising:

a verification system for authenticating digital signatures provided by the respective software

applications to access the APIs where the signature identifications correspond with the

signature identificaters of the respective APIs and where a digital signature for a software

application is generated with a signature identification corresponding to a signature identificater

to access at least one API (column 8, lines 56-65, column 9, lines 36-58, "an adapted digital

signature 144, is compared against the adapted digital signature in the incoming unique digital

signature" and column 10, lines 14-30, "media files are returned, such as Java applets, one or

more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

a control system for allowing access to at least one of the APIs where the digital signature

provided by the software application is authenticated by the verification system (column 5, lines

40-51, "an application server interface ('API') for messaging server 308 is added which provides

access to the authenticated message server services").

57.   (New) Regarding claim 113, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

58.   (New) Regarding claim 114, Gibbs discloses, wherein the virtual machine is a Java

virtual machine installed on a mobile device (column 6, lines 45-60, column 7, lines 2-8, "Java

applets", column 10, lines 14-30 and 35-45).

59.   (New) Regarding claim 115, Gibbs discloses, wherein the control system requires one

digital signature and one signature identification for each library of at least one of the APIs

(column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56,

column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-38 and 49-56 and column 11,

lines 8-12, "add-on software component in servers 424 or 408").

60.   (New) Regarding claim 116, Gibbs discloses, wherein the code signing system is

installed on a mobile device and the software application is a Java application for a mobile

device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

61.   (New) Regarding claim 117, Gibbs discloses, wherein the digital signature and the

signature identification of the software application are generated by a code signing authority

(column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column

8, lines 40-47).

62.   (New) Regarding claim 118, Gibbs discloses, wherein the APIs access *at least one of* a cryptographic module, which implements cryptographic algorithms, a data store, a proprietary data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32, column 9, lines 36-50 and column 10, lines 49-62).

63.   (New) Regarding claim 121, Gibbs discloses, wherein at least one of the APIs further comprises:

a description string that is displayed to a user when the software application attempts to access said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since remote users on legacy electronic message and existing telephone systems can still type the unique digital signature without special software (or hardware)" and column 10, lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62).

64.   (New) Regarding claim 122, Gibbs discloses, wherein the APIs provides access to at least one of one or more core functions of a mobile device, an operating system, and hardware on a mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines 8-13).

65.   (New) Regarding claim 123, Gibbs discloses, wherein verification of a global digital signature provided by the software application is required for accessing any of the APIs (Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

66.   (New) Regarding claim 124, Gibbs teaches a method of controlling access to application programming interfaces (APIs) having signature identifiers by software applications,

the method comprising:

authenticating digital signatures provided by the respective software applications to access the

APIs where the signature identifications correspond with the signature identifiers of the

respective APIs and where a digital signature for a software application is generated with a

signature identification corresponding to a signature identifier to access at least one API

(column 8, lines 56-65, column 9, lines 36-58, "an adapted digital signature 144, is compared

against the adapted digital signature in the incoming unique digital signature" and column 10,

lines 14-30, "media files are returned, such as Java applets, one or more bundled HTML files,

an MPEG file, a WAV file, or a RAM file" and lines 49-62);

allowing access to at least one of the APIs where the digital signature provided by the software

application is authenticated (column 5, lines 40-51, "an application server interface ('API') for

messaging server 308 is added which provides access to the authenticated message server

services").

67.   (New) Regarding claim 125, Gibbs teaches, wherein one digital signature and one

signature identification are provided by the software application access a library of at least one

of the APIs (column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and

40-56, column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-30 and 35-49 and

column 11, lines 8-12, "add-on software component in servers 424 or 408").

68.   (New) Regarding claim 126, Gibbs teaches, wherein the digital signature and the

signature identification of the software application are generated by a code signing authority

(column 3, lines 8-18, 25-40 and 64-67, column 4, lines 1-8, column 5, lines 45-51 and column

8, lines 40-47).

69.   (New) Regarding claim 127, Gibbs teaches, wherein the APIs access *at least one of* a

cryptographic module that implements cryptographic algorithms, a data store, a proprietary data

model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

70.   (New) Regarding claim 130, Gibbs teaches, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

71.   (New) Regarding claim 131, Gibbs teaches, wherein the APIs provides access to at

least one of one or more core functions of a mobile device, an operating system, and hardware

on a mobile device (column 6, lines 45-60, "laptop", column 8, lines 48-55 and column 10, lines

8-13).

72.   (New) Regarding claim 132, Gibbs teaches, wherein verification of a global digital

signature provided by the software application is required for accessing any of the APIs (Figure

6, column 8, lines 56-65 and column 10, lines 14-30).

73.   (New) Regarding claim 133, Gibbs discloses a management system for controlling

access by software applications to application programming interfaces (APIs) having at least

one signature identifier on a subset of a plurality of mobile devices, the management system

comprising:

a code signing authority for providing digital signatures and signature identifications to software

applications that require access to at least one of the APIs with a signature identifier on the

subset of the plurality of mobile devices, where a digital signature for a software application is

generated with a signature identification corresponding to a signature identifier, and the

signature identifications provided to the software applications comprise those signature

identifications that correspond to the signature identifiers that are substantially only on the

subset of the plurality of mobile devices (column 6, lines 45-60, "laptop", column 8, lines 48-55

and column 10, lines 8-13);

wherein each mobile device of the subset of the plurality of mobile devices comprises a

verification system for authenticating digital signatures provided by the respective software

applications to access respective APIs where the digital identifications correspond to the digital

identifiers of the respective APIs (column 8, lines 56-65, column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming unique

digital signature" and column 10, lines 14-30 and 49-62);

a control system for allowing the respective software applications to access at least one of the

APIs where the digital signatures provided by the respective software applications are

authenticated by the verification system (column 8, lines 56-67, column 9, lines 1-5 and 45-50,

column 10, lines 8-13, "prompts the remote user for access to any unique digital signatures

stored in a cookie file on laptop 452" and lines 31-67, "electronic voting or polling system. In

such a system, the unique digital signature 132 is allocated by the authenticated message

server 428 at step 704" and column 11, lines 1-7).

74.   (New) Regarding claim 134, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

75.   (New) Regarding claim 135, Gibbs discloses, wherein the virtual machine is a Java

virtual machine and the software applications are Java applications (column 6, lines 45-60,

column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

76.   (New) Regarding claim 136, Gibbs discloses, wherein the control system requires one

digital signature and one signature identification for each library of at least one of the APIs

(column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56,

column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12,

"add-on software component in servers 424 or 408").

77.   (New) Regarding claim 137, Gibbs discloses, wherein the APIs access at least one of a

cryptographic module, which implements cryptographic algorithms, a data store, a proprietary

data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

78.   (New) Regarding claim 140, Gibbs discloses, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

79.   (New) Regarding claim 141, Gibbs discloses, wherein the subset of the plurality of

mobile devices comprises mobile devices under the control of at least one of a corporation and

a carrier (column 2, lines 8-16, "subscribers of a particular service" and column 5, lines 52-65).

80.   (New) Regarding claim 142, Gibbs discloses, wherein a global digital signature provided

by the software application has to be authenticated before the software application is allowed

access to any of the APIs on a mobile device of the subset of the plurality of mobile devices

(Figure 6, column 8, lines 56-65 and column 10, lines 14-30).

81.   (New) Regarding claim 143, Gibbs teaches a method of controlling access by software

applications to application programming interfaces (APIs) having at least one signature identifier

on a subset of a plurality of mobile devices, the method comprising:

generating digital signatures for software applications with signature identifications

corresponding to respective signature identifiers of the APIs (column 3, lines 8-18, 25-40 and

64-67, column 4, lines 1-8, column 5, lines 41-51 and column 8, lines 40-47);

and providing the digital signatures and the signature identifications to software applications that

require access to at least one of the APIs on the subset of the plurality of mobile devices, where

the signature identifications provided to the software applications comprise those signature

identifications that correspond to the signature identifiers that are substantially only on the

subset of the plurality of mobile devices (column 4, lines 19-42, "log file", "100 records" and 50-

65, column 5, lines 11-19 and 40-56, column 6, lines 45-60, column 8, lines 2-10, column 10,

lines 14-30 and 35-49 and column 11, lines 8-12, "add-on software component in servers 424 or

408");

wherein each mobile device of the subset of the plurality of mobile devices comprises

a verification system for authenticating digital signatures provided by the respective software

applications to access respective APIs where the digital identifications correspond to the digital

identifiers of the respective APIs (column 8, lines 56-65, column 9, lines 36-58, "an adapted

digital signature 144, is compared against the adapted digital signature in the incoming unique

digital signature" and column 10, lines 14-30, "media files are returned, such as Java applets,

one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file" and lines 49-62);

a control system for allowing the software application to access at least one of the APIs where

the digital signature provided by the software application is authenticated by the verification

system (column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts

the remote user for access to any unique digital signatures stored in a cookie file on laptop 452"

and lines 31-67, "electronic voting or polling system. In such a system, the unique digital

signature 132 is allocated by the authenticated message server 428 at step 704" and column

11, lines 1-7).

82.   (New) Regarding claim 144, Gibbs teaches, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

83.   (New) Regarding claim 145, Gibbs teaches, wherein the virtual machine is a Java virtual

machine and the software applications are Java applications (column 6, lines 45-60, column 7,

lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

84.   (New) Regarding claim 146, Gibbs teaches, wherein the control system requires one

digital signature and one signature identification for each library of at least one of the APIs

(column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56,

column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12,

"add-on software component in servers 424 or 408").

85.   (New) Regarding claim 147, Gibbs teaches, wherein the APIs access at least one of a

cryptographic module, which implements cryptographic algorithms, a data store, a proprietary

data model, and a user interface (UI) (column 3, lines 10-18, "an authenticated message server

functionally comprises a digital service engine 120", column 5, lines 29-51, "Authenticated

message server 316 can run on a standard personal computer" and lines 60-65, column 6, lines

30-39 and 61-66, column 7, lines 54-58, "user's personal computer", column 8, lines 11-32,

column 9, lines 36-50 and column 10, lines 49-62).

86.   (New) Regarding claim 150, Gibbs teaches, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

87.   (New) Regarding claim 151, Gibbs teaches, wherein the subset of the plurality of mobile

devices comprises mobile devices under the control of at least one of a corporation and a

carrier (column 2, lines 8-16, "subscribers of a particular service" and column 5, lines 52-65).

88.   (New) Regarding claim 152, Gibbs teaches a mobile device for a subset of a plurality of

mobile devices, the mobile device comprising:

an application platform having application programming interfaces (APIs) (column 6, lines 45-

60, "laptop", column 8, lines 48-55 and column 10, lines 8-13);

a verification system for authenticating digital signatures and signature identifications provided

by the respective software applications to access the APIs (column 8, lines 56-65, column 9,

lines 36-58, "an adapted digital signature 144, is compared against the adapted digital signature

in the incoming unique digital signature" and column 10, lines 14-30, "media files are returned,

such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file"

and lines 49-62);

a control system for allowing a software application to access at least one of the APIs where a

digital signature provided by the software application is authenticated by the verification system

(column 8, lines 56-67, column 9, lines 1-5 and 45-50, column 10, lines 8-13, "prompts the

remote user for access to any unique digital signatures stored in a cookie file on laptop 452" and

lines 31-67, "electronic voting or polling system. In such a system, the unique digital signature

132 is allocated by the authenticated message server 428 at step 704" and column 11, lines 1-

7);

wherein a code signing authority provides digital signatures and signature identifications to

software applications that require access to at least one of the APIs such that the digital

signature for the software application is generated according to a signature scheme of a

signature identification, and wherein the signature identifications provided to the software

applications comprise those signature identifications that are substantially only authorized to

allow access on the subset of the plurality of mobile devices (column 3, lines 8-18, 25-40 and

64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

89.   (New) Regarding claim 153, Gibbs discloses, wherein a virtual machine comprises the

verification system and the control system (Figure 6, column 8, lines 56-65 and column 10, lines

14-30).

90.   154. (New) Regarding claim 154, Gibbs discloses, wherein the virtual machine is a Java

virtual machine and the software application is a Java application (column 6, lines 45-60,

column 7, lines 2-8, "Java applets", column 10, lines 14-30 and 35-45).

91.   (New) Regarding claim 155, Gibbs discloses, wherein the control system requires one

digital signature and one signature identification for each library of at least one of the APIs

(column 4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56,

column 6, lines 45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12,

"add-on software component in servers 424 or 408").

92.   (New) Regarding claim 156, Gibbs discloses, wherein the APIs of the application

platform access at least one of a cryptographic module, which implements cryptographic

algorithms, a data store, a proprietary data model, and a user interface (UI) (column 3, lines 10-

18, "an authenticated message server functionally comprises a digital service engine 120",

column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal

computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's

personal computer", column 8, lines 11-32, column 9, lines 36-50 and column 10, lines 49-62).

93.   (New) Regarding claim 159, Gibbs discloses, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to access

said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is used since

remote users on legacy electronic message and existing telephone systems can still type the

unique digital signature without special software (or hardware)" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62).

94.   (New) Regarding claim 160, Gibbs teaches a method of controlling access to

application programming interfaces (APIs) of an application platform of a mobile device for a

subset of a plurality of mobile devices, the method comprising:

receiving digital signatures and signature identifications from software applications that require

to access the APIs authenticating the digital signatures and the signature identifications (column

8, lines 56-65, column 9, lines 36-58, "an adapted digital signature 144, is compared against the

adapted digital signature in the incoming unique digital signature" and column 10, lines 14-30,

"media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file,

a WAV file, or a RAM file" and lines 49-62);

allowing a software application to access at least one of the APIs where a digital signature

provided by the software application is authenticated (column 8, lines 56-67, column 9, lines 1-5

and 45-50, column 10, lines 8-13, "prompts the remote user for access to any unique digital

signatures stored in a cookie file on laptop 452" and lines 31-67, "electronic voting or polling

system. In such a system, the unique digital signature 132 is allocated by the authenticated

message server 428 at step 704" and column 11, lines 1-7);

wherein a code signing authority provides the digital signatures and the signature identifications

to the software applications that require access to at least one of the APIs such that the digital

signature for the software application is generated according to a signature scheme of a

signature identification, and wherein the signature identifications provided to the software

applications comprise those signature identifications that are substantially only authorized to

allow access on the subset of the plurality of mobile devices (column 3, lines 8-18, 25-40 and

64-67, column 4, lines 1-8, column 5, lines 45-51 and column 8, lines 40-47).

95.   (New) Regarding claim 161, Gibbs teaches, wherein one digital signature and one

signature identification is required for accessing each library of at least one of the APIs (column

4, lines 19-42, "log file", "100 records" and 50-65, column 5, lines 11-19 and 40-56, column 6,

lines 45-60, column 8, lines 2-10, column 10, lines 14-38 and column 11, lines 8-12, "add-on

software component in servers 424 or 408").

96.   162. (New) Regarding claim 162, Gibbs teaches, wherein the APIs of the application

platform access at least one of a cryptographic module, which implements cryptographic

algorithms, a data store, a proprietary data model, and a user interface (UI) (column 3, lines 10-

18, "an authenticated message server functionally comprises a digital service engine 120",

column 5, lines 29-51, "Authenticated message server 316 can run on a standard personal

computer" and lines 60-65, column 6, lines 30-39 and 61-66, column 7, lines 54-58, "user's

personal computer", column 8, lines 11-32, column 9, lines 36-50 and column 10, lines 49-62).

97.   (New) Regarding claim 165, Gibbs teaches, wherein at least one of the APIs further

comprises:

a description string that is displayed to a user when the software application attempts to

access said at least one of the APIs (column 8, lines 28-39, "a limited character ASCII set is

used since remote users on legacy electronic message and existing telephone systems can still

type the unique digital signature without special software (or hardware)" and column 10, lines

14-30, "media files are returned, such as Java applets, one or more bundled HTML files, an

MPEG file, a WAV file, or a RAM file" and lines 49-62).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject
> matter sought to be patented and the prior art are such that the subject matter as a
> whole would have been obvious at the time the invention was made to a person having
> ordinary skill in the art to which said subject matter pertains. Patentability shall not be
> negatived by the manner in which the invention was made.

> The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459

(1966), that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

> 1.   Determining the scope and contents of the prior art.
> 2.   Ascertaining the differences between the prior art and the claims at issue.
> 3.   Resolving the level of ordinary skill in the pertinent art.
> 4.   Considering objective evidence present in the application indicating obviousness
>   or nonobviousness.

98.   Claim 78 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gibbs as

applied to claim 57 above, and further in view of United States Patent No. 6,584,376 to Van

Kommer, hereinafter Van Kommer.

99.   Gibbs significantly discloses the claimed invention, as cited above. However, Gibbs

does not substantially disclose the claim language of claim 78 pertaining to "wherein the

hardware comprises a subscriber identity module (SIM) card". Van Kommer discloses this, as

cited below.

100.   (New) Regarding claim 78, Van Kommer discloses, wherein the hardware comprises a

subscriber identity module (SIM) card (column 3, lines 22-33, "mobile phone comprises

preferably a subscriber identification module 300, for example a removable SIM card, which

enables the mobile robot to be identified within the mobile telecommunications network 2" and

column 5, lines 31-36).

101.   The motivation to combine would be to provide a means for "the mobile robot to be

identified within the mobile telecommunications network" (*Van Kommer* – column 3, lines 24-

27).

102.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to incorporate the teachings of Van Kommer with the teachings of Gibbs in

order "to verify the identity and privileges of the distant operator 1, for example a mobile

subscriber" (*Van Kommer* - column 6, lines 56-64).

103.   Claims 72, 73, 88, 89, 98, 109, 110, 119, 120, 128, 129, 138, 139, 148, 149, 157, 158,

163 and 164 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gibbs as applied

to claims 57, 83, 92, 103, 112, 124, 133, 143, 152 and 160 above, and further in view of United

States Patent No. 6,587,837 to Spagna et al., hereinafter Spagna.

104.   Gibbs significantly discloses the claimed invention, as cited above. However, Gibbs

does not substantially disclose the claim language pertaining to the "public signature key" and

"private signature key" as found within the following claims. Spagna discloses this claim

language, as cited below.

105.   Based upon the similarities of the claim language between the following claims, the

following motivation and obviousness to combine is applicable to each of the subsequent

claims.

106.   The motivation to combine would be that "the issuer of SC(s) protects the integrity of

SC(s) by digitally signing it" (*Spagna* - column 17, lines 1-14).

107.   Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to incorporate the teachings of Spagna with the teachings of Gibbs so that

"it is easy to calculate the digest for an input message, but it is very difficult (computationally

infeasible) to generate "he input message from its digest" (*Spagna* – column 17, lines 25-31).

108.   (New) Regarding claim 72, Spagna discloses, wherein the digital signature is generated

using a private signature key, and the virtual machine uses a public signature key to verify the

authenticity of the digital signature (column 16, lines 46-53, "Public key algorithms are also used

to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14

and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the

signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and

column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and

decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity

included if signed").

109.   (New) Regarding claim 73, Spagna discloses, wherein:

the digital signature is generated by applying the private signature key to a hash of the software

application (column 17, lines 25-33, column 27, lines 41-51);

the virtual machine verifies the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

110.   (New) Regarding claim 88, Spagna teaches, wherein the digital signature is generated by applying a private signature key to a hash of the software application (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed"),

and wherein the step of verifying the authenticity of the digital signature is performed by a method comprising the steps of:

storing a public signature key that corresponds to the private signature key on the mobile device (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed");

generating a hash of the software application to obtain a generated hash (column 17, lines 25-33, column 27, lines 41-51);

applying the public signature key to the digital signature to obtain a recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49);

comparing the generated hash with the recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

111.   (New) Regarding claim 89, Spagna teaches, wherein the digital signature is generated by calculating a hash of the software application and applying the private signature key (column 17, lines 25-33, column 27, lines 41-51).

112.   (New) Regarding claim 98, Spagna teaches, wherein the signature key is a private signature key and the corresponding signature key is a public signature key (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

113.   (New) Regarding claim 109, Spagna teaches, wherein each of the plurality of digital signatures and signature identifications are generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to

create the signature thus in EU is the End-User(s) digital signature", column 18, lines 46-60,

column 19, lines 19-30, column 27, lines 41-51, column 36, lines 9-19 and 43-49 and column

46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts

the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if

signed").

114.   (New) Regarding claim 110, Spagna teaches, wherein the step of authenticating the

digital signature where the signature identification corresponds with the signature identifier

comprises the steps of:

verifying that the signature identification corresponds with the signature identifier authenticating

the digital signature where signature identification corresponds with the signature identifier

comprising the steps of:

storing a public signature key on a mobile device that corresponds to the private signature key

associated with the code signing authority which generates the digital signature (column 16,

lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key

is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials

indicate which private key was used to create the signature thus in EU is the End-User(s) digital

signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed");

generating a hash of the software application to obtain a generated hash (column 17, lines 25-

33, column 27, lines 41-51);

applying the public signature key to the digital signature to obtain a recovered hash (column 18,

lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-

49);

and comparing the generated hash with the recovered hash (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

115.   (New) Regarding claim 119, Spagna discloses, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and the verification system uses a public signature key to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

116.   (New) Regarding claims 120, 139, 158 and 164, Spagna discloses wherein:
the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme (column 17, lines 25-33, column 27, lines 41-51);
the verification system authenticates the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

117.   (New) Regarding claim 128, Spagna teaches, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and a public signature key is used to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials

indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

118. (New) Regarding claim 129, Spagna teaches, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme (column 17, lines 25-33, column 27, lines 41-51);

the digital signature is authenticated by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and verifying that the generated hash with the recovered hash are the same (column 18, lines 46-60, column 19, lines 19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

119. (New) Regarding claim 138, Spagna discloses, wherein the digital signature is generated using a private signature key under a signature scheme associated with the signature identification, and the verification system uses a public signature key to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

120. (New) Regarding claim 148, Spagna teaches, wherein at least one of the digital signatures is generated using a private signature key under a signature scheme associated with a signature identification, and the verification system uses a public signature keys to

authenticate said at least one of the digital signatures (column 16, lines 46-53, "Public key

algorithms are also used to generate digital signatures. The private key is used for that

purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which

private key was used to create the signature thus in EU is the End-User(s) digital signature",

column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself

using the Public Key 661 of the signing entity included if signed").

121.   (New) Regarding claim 149, Spagna teaches, wherein:

at least one of the digital signatures is generated by applying the private signature key to a hash

of a software application under the signature scheme (column 17, lines 25-33, column 27, lines

41-51);

the verification system authenticates said at least one of the digital signatures by generating a

hash of the software application to obtain a generated hash, applying the public signature key to

said at least one of the digital signatures to obtain a recovered hash, and verifying that the

generated hash with the recovered hash are the same (column 18, lines 46-60, column 19, lines

19-30, column 27, lines 41-51 and column 36, lines 9-19 and 43-49).

122.   (New) Regarding claim 157, Spagna discloses, wherein the digital signature is

generated using a private signature key under the signature scheme, and the verification

system uses a public signature key to authenticate the digital signature (column 16, lines 46-53,

"Public key algorithms are also used to generate digital signatures. The private key is used for

that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which

private key was used to create the signature thus in EU is the End-User(s) digital signature",

column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital

signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

123.   (New) Regarding claim 163, Spagna teaches, wherein the digital signature is generated using a private signature key under the signature scheme, and a public signature key is used to authenticate the digital signature (column 16, lines 46-53, "Public key algorithms are also used to generate digital signatures. The private key is used for that purpose.", column 17, lines 2-14 and 38-46, column 18, lines 2-11, "initials indicate which private key was used to create the signature thus in EU is the End-User(s) digital signature", column 36, lines 10-19 and 43-49 and column 46, lines 40-58, "To validate the digital signatures, first the Clearinghouse(s) 105 and decrypts the Contents 631 of the signature itself using the Public Key 661 of the signing entity included if signed").

Conclusion

124.   The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The following United States Patents are cited to further show the state of the art with respect to ensuring the security of data, such as:

United States Patent No. 6,574,609 to Downs, et al., which is cited to show a secure electronic content management system.

United States Patent No. 6,324,650 to Ogilvie, which is cited to show message content protection and conditional disclosure.

United States Patent No. 6,795,923 to Stern, et al., which is cited to show a mechanism for embedding network based control systems in a local network interface device.

United States Patent No. 7,243,236 to Silbert, which is cited to show systems and methods for using cryptography to protect secure and insecure computing environments.

125.   Any inquiry concerning this communication or earlier communications from the examiner

should be directed to JEREMIAH AVERY whose telephone number is (571)272-8627. The

examiner can normally be reached on Monday thru Friday 8:30am-5pm.

126.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Ayaz Sheikh can be reached on (571) 272-3795. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

127.   Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private

PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you

would like assistance from a USPTO Customer Service Representative or access to the

automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jeremiah Avery/
Examiner, Art Unit 2431
      /Ayaz R. Sheikh/
      Supervisory Patent Examiner, Art Unit 2431

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Notice of References Cited		10/381,219	YACH ET AL.
		Examiner	Art Unit	
		JEREMIAH AVERY	2431	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,795,919	09-2004	Gibbs et al.	713/170
*	B	US-6,587,837	07-2003	Spagna et al.	705/26
*	C	US-6,574,609	06-2003	Downs et al.	705/50
*	D	US-6,324,650	11-2001	Ogilvie, John W.L.	726/2
*	E	US-6,795,923	09-2004	Stern et al.	726/12
*	F	US-7,243,236	07-2007	Sibert, W. Olin	713/179
*	G	US-6,584,376	06-2003	Van Kommer, Robert	700/245
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

U.S. Patent and Trademark Office
PTO-892 (Rev. 01-2001)   **Notice of References Cited**   Part of Paper No. 20081113

	Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant	☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE									
Final	Original	03/12/2009	03/12/2009								
	1	-									
	2	-									
	3	-									
	4	-									
	5	-									
	6	-									
	7	-									
	8	-									
	9	-									
	10	-									
	11	-									
	12	-									
	13	-									
	14	-									
	15	-									
	16	-									
	17	-									
	18	-									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
	26	-									
	27	-									
	28	-									
	29	-									
	30	-									
	31	-									
	32	-									
	33	-									
	34	-									
	35	-									
	36	-									

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	**Art Unit**
	JEREMIAH AVERY	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

| ☐ Claims renumbered in the same order as presented by applicant | | ☐ CPA | ☐ T.D. | ☐ R.1.47 |

CLAIM		DATE							
Final	Original	03/12/2009	03/12/2009						
	37	-							
	38	-							
	39	-							
	40	-							
	41	-							
	42	-							
	43	-							
	44	-							
	45	-							
	46	-							
	47	-							
	48	-							
	49	-							
	50	-							
	51	-							
	52	-							
	53	-							
	54	-							
	55	-							
	56	-							
	57	✓							
	58	✓							
	59	✓							
	60	✓							
	61	✓							
	62	✓							
	63	✓							
	64	✓							
	65	✓							
	66	✓							
	67	✓							
	68	✓							
	69	✓							
	70	✓							
	71	✓							
	72	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10381219	YACH ET AL.
	Examiner	Art Unit
	JEREMIAH AVERY	2431

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE									
Final	Original	03/12/2009	03/12/2009								
	73	✓									
	74	✓									
	75	✓									
	76	✓									
	77	✓									
	78	✓									
	79	✓									
	80	✓									
	81	✓									
	82	✓									
	83	✓									
	84	✓									
	85	✓									
	86	✓	O								
	87	✓									
	88	✓									
	89	✓									
	90	✓									
	91	✓									
	92	✓	O								
	93	✓									
	94	✓									
	95	✓									
	96	✓									
	97	✓									
	98	✓									
	99	✓									
	100	✓									
	101	✓									
	102	✓									
	103	✓									
	104	✓									
	105	✓	O								
	106	✓									
	107	✓									
	108	✓									

Index of Claims		**Application/Control No.** 10381219		**Applicant(s)/Patent Under Reexamination** YACH ET AL.						
		Examiner JEREMIAH AVERY		**Art Unit** 2431						

✓	Rejected		-	Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE									
Final	Original	03/12/2009	03/12/2009								
	109	✓									
	110	✓									
	111	✓									
	112	✓	O								
	113	✓									
	114	✓									
	115	✓									
	116	✓									
	117	✓									
	118	✓									
	119	✓									
	120	✓									
	121	✓									
	122	✓									
	123	✓									
	124	✓									
	125	✓									
	126	✓									
	127	✓									
	128	✓									
	129	✓									
	130	✓									
	131	✓									
	132	✓	O								
	133	✓									
	134	✓									
	135	✓									
	136	✓									
	137	✓									
	138	✓									
	139	✓									
	140	✓									
	141	✓									
	142	✓									
	143	✓									
	144	✓									

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA	☐ T.D.	☐ R.1.47

CLAIM		DATE							
Final	Original	03/12/2009	03/12/2009						
	145	✓							
	146	✓							
	147	✓							
	148	✓							
	149	✓							
	150	✓							
	151	✓							
	152	✓							
	153	✓							
	154	✓							
	155	✓							
	156	✓							
	157	✓							
	158	✓							
	159	✓							
	160	✓							
	161	✓							
	162	✓							
	163	✓							
	164	✓							
	165	✓							

Page 978 of 1415

Search Notes	**Application/Control No.** 10381219	**Applicant(s)/Patent Under Reexamination** YACH ET AL.
	Examiner JEREMIAH AVERY	**Art Unit** 2431

SEARCHED

Class	Subclass	Date	Examiner
none	none	3/12/2009	JLA

SEARCH NOTES

Search Notes	Date	Examiner
Inventor Search	3/6/2009	JLA
Keywords in EAST Search	3/6/2009	JLA

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
none	none	3/12/2009	JLA

UNITED STATES PATENT AND TRADEMARK OFFICE

BIB DATA SHEET

CONFIRMATION NO. 9761

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/381,219	03/20/2003 RULE	713	2431	555255012423

APPLICANTS
David P Yach, Waterloo, ON, CANADA;
Michael S Brown, Waterloo, ON, CANADA;
Herbert A Little, Waterloo, ON, CANADA;

** CONTINUING DATA **************************
This application is a 371 of PCT/CA01/01344 09/20/2001
    which claims benefit of 60/234,152 09/21/2000
    and claims benefit of 60/235,354 09/26/2000
    and claims benefit of 60/270,663 02/20/2001

** FOREIGN APPLICATIONS *************************

** IF REQUIRED, FOREIGN FILING LICENSE GRANTED **
02/27/2004

Foreign Priority claimed ☑Yes ☐No		STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
35 USC 119(a-d) conditions met ☑Yes ☐No	☐ Met after Allowance				
Verified and Acknowledged /JEREMIAH L AVERY/ Examiner's Signature	JLA Initials	ON	7	109	12

ADDRESS

David B Cochran
Jones Day
North Point
901 Lakeside Avenue
Cleveland, OH 44114-1190
UNITED STATES

TITLE

Software code signing system and method

FILING FEE RECEIVED 3258	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees
		☐ 1.16 Fees (Filing)
		☐ 1.17 Fees (Processing Ext. of time)
		☐ 1.18 Fees (Issue)
		☐ Other _____
		☐ Credit

BIB (Rev. 05/07).

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	David P. Yach, et al.
Serial No.:	10/381,219
Filed:	March 20, 2003
For:	SOFTWARE CODE SIGNING SYSTEM AND METHOD
Art Unit:	2131
Examiner:	Avery, Jeremiah L.

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure imposed by 37 C.F.R. § 1.56, applicants hereby advise the United States Patent and Trademark Office of certain references which may be material to the determination of patentability of the above-identified application. The references are identified on the attached Form PTO-1449 and copies of the references are enclosed, if required. Applicants respectfully request that these references be considered and made of record in the present application by completing and returning the enclosed Form PTO-1449.

No fee is believed to be due for entry of this Information Disclosure Statement. However, if any fee should be required, please charge such fee to Jones Day's Deposit Account No. 501432, Reference No. 555255-012423.

Respectfully submitted,

David B. Cochran
Reg. No. 39,142
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939

I hereby certify that this correspondence is being deposited today with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450,

on _____Feb 1, 2007_____

By: _____

Page 1 of 1

CLI-114663v1

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	10/381,219
Filing Date	March 20, 2003
First Named Inventor	David P. Yach, et al
Art Unit	~~2134~~ 2431
Examiner Name	Avery, Jeremiah L.

Sheet	1	of	1	Attorney Docket Number	555255-012423

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Communication of Notices of Opposition (R. 57(1) EPC) dated 26-09-2006 and Working Translation, 16 pages	
		ISO/IEC FCD 7816-9 "Identification cards ...", Part 9: Additional interindustry commands and security attributes", 17.06.1999, S. 8 bis 13, 29 bis 31 (D5), 12 pages	
		ISO/IEC FDIS 7816-8 "Identification cards ...", Part 8: Security related interindustry commands", 25.06.1998, S. 2, 3, 6 bis 13 (D6), 13 pages	
		ISO/IEC 7816-4 "Information Technology - Identification Cards...", Part 4: Interindustry Commands for Interchange", 1995, S. 12 bis 16 (D7), 6 pages	
~~|	Handbuch der Chipkarten, W. Rankl/W. Effing, 3. Auflage Hanser-Verlag Munchen, 1999, S. 197 bis 203, 261 bis 272, 740, 795 bis 797 (D8), 18 pages |~~	

Examiner Signature	/Jeremiah Avery/	Date Considered	02/19/2009

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

PATENT

Attorney Docket No. 555255012423

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	David P. Yach, et al.
Serial No.:	10/381,219
Filed:	March 20, 2003
For:	CODE SIGNING SYSTEM AND METHOD
Art Unit:	2131
Examiner:	Not yet assigned

RECEIVED

DEC 1 7 2003
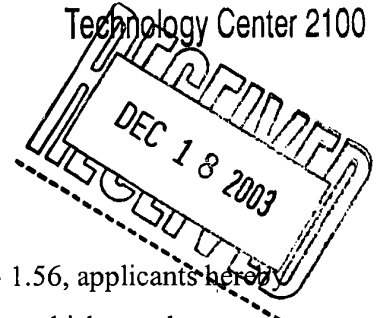
Technology Center 2100

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure imposed by 37 C.F.R. § 1.56, applicants hereby advise the United States Patent and Trademark Office of certain references which may be material to the determination of patentability of the above-identified application. The references are identified on the attached Form PTO-1449 and copies of the references are enclosed. Applicants respectfully request that these references be considered and made of record in the present application by completing and returning the enclosed Form PTO-1449.

No fee is believed to be due for entry of this Information Disclosure Statement. However, if any fee should be required, please charge such fee to Jones Day's Deposit Account No. 501432, Reference No. 555255012423.

Respectfully submitted,

I hereby certify that this correspondence is being deposited today with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents. P.O. Box 1450. Alexandria. VA 22313-1450

on _____Dec. 11, 2003_____

By: _____

David B. Cochran
Reg. No. 39,142
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939

Page 1 of 1

CLI-1146691v1 ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

FORM PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE **INFORMATION DISCLOSURE STATEMENT BY APPLICANT** (Use several sheets if necessary) (37 CFR 1.98(b))	Atty Docket No.: 555255012423
	Application No.: 10/381,219
	Applicants: David P. Yach, et al.
	Filing Date: March 20, 2003
	Group: ~~2131~~ 2431
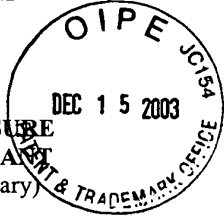
Stamp: OIPE JC154 DEC 1 5 2003 PATENT & TRADEMARK OFFICE

Stamp: RECEIVED DEC 1 7 2003 Technology Center 2100

U.S. PATENT (AND PATENT PUBLICATION) DOCUMENTS

Exam. Init.		Document No.							Date MM/DD/YYYY	Name	Class	Subclass	Filing Date
	AA	5	9	7	8	4	8	4	11/02/1999	Apperson et al.			
	AB	6	1	5	7	7	2	1	12/05/2000	Shear et al.			
	AC												
	AD												
	AE												
	AF												
	AG												
	AH												
	AI												
	AJ												
	AK												
	AL												
	AM												

FOREIGN PATENT OR PUBLISHED FOREIGN PATENT APPLICATION

Exam. Init.		Document Number							Publication Date of the Grant	Country or Patent Office	Class	Subclass	Translation Yes	Translation No
	AN	9	9	0	5	6	0	0	02/04/1999	WO				
	AO	0	9	3	0	7	9	3	07/21/1999	EP				
	AP													
	AQ													
	AR													

OTHER DOCUMENTS (Including Author, Title, Date**, Relevant pages, Place of Publication***)

	AS	
	AT	
	AU	

Examiner /Jeremiah Avery/	Date Considered 02/19/2009

EXAMINER: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

CLI-1146628v1   ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	8	(digital near signature) and (authentic$ or verify$ or verificat$) and (virtual near machine) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:23
S2	8	S1 and (portab$ or mobile or handheld or laptop or pda or cell or cellular)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S3	4	S2 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 11:24
S4	35	(digital near signature) and (authentic$ or verify$ or verificat$) and (java or (virtual near machine)) and ((API) or (Application near programming near interface)) and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5) same (digital near signature)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:28
S5	737	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33

file:///C|/Documents%20and%20Settings/javery/My%20Doc...1219/EASTSearchHistory.10381219_AccessibleVersion.htm (1 of 5)3/12/2009 6:08:08 PM

Page 985 of 1415

S6	41	S5 and ((deny$ or denies or denial) same (digital near signature)) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear $5 or remov$5 or revok$ or revocat$) same (digital near signature))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:33
S7	30	S6 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:37
S8	30	S7 and access$	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S9	30	S8 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:38
S10	2	S9 and ((secure near hash near algorithm) or SHA?1)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:39
S11	1	S10 and public and private	US-PGPUB; USPAT	OR	ON	2009/02/20 11:40
S12	31	S6 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:46
S13	31	S12 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:47
S14	31	S13 and ((public or private) same key)	US-PGPUB; USPAT	OR	ON	2009/02/20 11:48
S15	0	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) same (((secure near hash near algorithm) or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53

file:///C|/Documents%20and%20Settings/javery/My%20Doc...1219/EASTSearchHistory.10381219_AccessibleVersion.htm (2 of 5)3/12/2009 6:08:08 PM

Page 986 of 1415

S16	30	S14 and (((portable or portability or mobile or handheld or cell or cellular) near phone) or laptop or pda) and (((secure near hash near algorithm) or (SHA1 of SHA? 1)) or (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 11:53
S17	28	S16 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:07
S18	118	S5 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S19	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:25
S20	56	S19 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:26
S21	61	S18 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S22	56	S21 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/02/20 12:27
S23	40	S22 and wireless	US-PGPUB; USPAT	OR	ON	2009/02/20 12:28
S24	55	S22 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/02/20 12:32

file:///C|/Documents%20and%20Settings/javery/My%20Doc...1219/EASTSearchHistory.10381219_AccessibleVersion.htm (3 of 5)3/12/2009 6:08:08 PM

Page 987 of 1415

S25	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S26	119	S25 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S27	62	S26 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S28	16	S27 and (SIM or (subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:09
S29	30	S25 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda) and (SIM or (Subscriber near identi$ near module))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:14
S30	16	S29 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:24
S31	9	S28 and (display or visual) and (API or (application near program$ near interface))	US-PGPUB; USPAT	OR	ON	2009/03/05 15:28
S32	738	(digital near signature) and (authentic$ or verify$ or verificat$ or validat$ or valid) and (java or (virtual near machine)) and (@ad<"20000921" @prad<"20000921")	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51

file:///C|/Documents%20and%20Settings/javery/My%20Doc...1219/EASTSearchHistory.10381219_AccessibleVersion.htm (4 of 5)3/12/2009 6:08:08 PM

Page 988 of 1415

S33	119	S32 and ((deny$ or denies or denial) same access$) and ((eras$ or purg$ or delet$ or expung$ or eliminat$ or eradicat$ or clear$5 or remov $5 or revok$ or revocat$) same (software or program or application))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S34	62	S33 and (((portable or portability or mobile or handheld or cell or cellular) near (device or computer or apparatus or phone)) or laptop or pda)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S35	57	S34 and ((secure near hash near algorithm or (SHA1 of SHA?1)) or (hash$ or (one? way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S36	56	S35 and ((public or private) near key)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S37	56	S36 and (hash$ or (one?way or (one near way)))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:51
S38	36	S37 and (digital near signature) and ((authentic$ or verify$ or verification) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S39	0	S38 and (signature near (hash $ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:52
S40	3	S38 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:53
S41	40	S37 and (digital near signature) and ((authentic$ or verify$ or verification or valid $) near signature)	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57
S42	6	S41 and (signature same (hash$ or (one?way or (one near way))))	US-PGPUB; USPAT	OR	ON	2009/03/06 16:57

3/12/2009 6:07:55 PM
C:\ Documents and Settings\ javery\ My Documents\ EAST\ Workspaces\ 10381219.wsp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	David P. Yach; Michael S. Brown; Herbert A. Little
Internat'l. Appl'n. No.	:	PCT/CA01/01344
Internat'l. Filing Date	:	09/20/2001
U.S. Serial No.	:	10/381,219
U.S. Filing Date	:	03/20/2003
Priority Date Claimed:		09/21/2000
Title	:	Software Code Signing System And Method
Art Unit	:	2131
Examiner	:	J. Avery
Docket No.	:	555255012423

Commissioner for Patents
Washington, D.C. 20231

Preliminary Amendment

This paper responds to the notice of non-compliant amendment mailed May 21, 2007.
Any fees due should be charged to Jones Day Deposit Account No. 501432, ref: 555255-012423.

Prior to taking up this case for initial examination, please amend the application as follows.

Page 990 of 1415

The Claims

Please <u>cancel</u> original claims 1-56.

Please <u>add</u> the following new claims 57-165.

57. (New) A code signing system for operation in conjunction with a software application having a digital signature and a signature identification, where the digital signature is associated with the signature identification, comprising:

   an application platform;

   an application programming interface (API) having an associated signature identifier, the API is configured to link the software application with the application platform; and

   a virtual machine that verifies the authenticity of the digital signature in order to control access to the API by the software application where the signature identifier corresponds to the signature identification.

58. (New) The code signing system of claim 57, wherein the virtual machine denies the software application access to the API if the digital signature is not authenticated.

59. (New) The code signing system of claim 57, wherein the virtual machine purges the software application if the digital signature is not authenticated.

60. (New) The code signing system of claim 57, wherein the code signing system is installed on a mobile device.

61. (New) The code signing system of claim 57, wherein the digital signature is generated by a code signing authority.

62. (New) A code signing system for operation in conjunction with a software application having a digital signature and a signature identification where the digital signature is associated with the signature identification, comprising:

   an application platform;

a plurality of application programming interfaces (APIs) associated with a signature identifier, each configured to link the software application with a resource on the application platform; and

a virtual machine that verifies the authenticity of the digital signature in order to control access to the APIs by the software application where the signature identification corresponds to the signature identifier,

wherein the virtual machine verifies the authenticity of the digital signature in order to control access to the plurality of APIs by the software application.

63.   (New) The code signing system of claim 62, wherein the plurality of APIs are included in an API library.

64.   (New) The code signing system of claim 62, wherein one or more of the plurality of APIs is classified as sensitive and having an associated signature identifier, and wherein the virtual machine uses the digital signature and the signature identification to control access to the sensitive APIs.

65.   (New) The code signing system of claim 64, wherein the code signing system operates in conjunction with a plurality of software applications, wherein one or more of the plurality of software applications has a digital signature and a signature identification, and wherein the virtual machine verifies the authenticity of the digital signature of each of the one or more of the plurality of software applications, where the signature identification corresponds to the signature identifier of the respective sensitive APIs, in order to control access to the sensitive APIs by each of the plurality of software applications.

66.   (New) The code signing system of claim 62, wherein the resource on the application platform comprises a wireless communication system.

67.   (New) The code signing system of claim 62, wherein the resource on the application platform comprises a cryptographic module which implements cryptographic algorithms.

68.   (New) The code signing system of claim 62, wherein the resource on the application platform comprises a data store.

69.   (New) The code signing system of claim 62, wherein the resource on the application platform comprises a user interface (UI).

70.   (New) The code signing system of claim 57, further comprising:
a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs, wherein the virtual machine controls access to the plurality of API libraries by the software application.

71.   (New) The code signing system of claim 70, wherein at least one of the plurality of API libraries is classified as sensitive;
wherein access to a sensitive API library requires a digital signature associated with a signature identification where the signature identification corresponds to a signature identifier associated with the sensitive API library;
wherein the software application includes at least one digital signature and at least one associated signature identification for accessing sensitive API libraries; and
wherein the virtual machine authenticates the software application for accessing the sensitive API library by verifying the one digital signature included in the software application that has a signature identification corresponding to the signature identifier of the sensitive API library.

72.   (New) The code signing system of claim 57, wherein the digital signature is generated using a private signature key, and the virtual machine uses a public signature key to verify the authenticity of the digital signature.

73.   (New) The code signing system of claim 72, wherein:
the digital signature is generated by applying the private signature key to a hash of the software application; and

the virtual machine verifies the authenticity of the digital signature by generating a hash of the software application to obtain a generated hash, applying the public signature key to the digital signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

74.   (New) The code signing system of claim 60, wherein the API further comprises:
a description string that is displayed by the mobile device when the software application attempts to access the API.

75.   (New) The code signing system of claim 57, wherein the application platform comprises an operating system.

76   (New) The code signing system of claim 57, wherein the application platform comprises one or more core functions of a mobile device.

77.   (New) The code signing system of claim 57, wherein the application platform comprises hardware on a mobile device.

78.   (New) The code signing system of claim 57, wherein the hardware comprises a subscriber identity module (SIM) card.

79.   (New) The code signing system of claim 57, wherein the software application is a Java application for a mobile device.

80.   (New) The code signing system of claim 57, wherein the API interfaces with a cryptographic routine on the application platform.

81.   (New) The code signing system of claim 57, wherein the API interfaces with a proprietary data model on the application platform.

82. (New) The code signing system of claim 57, wherein the virtual machine is a Java virtual machine installed on a mobile device.

83. (New) A method of controlling access to sensitive application programming interfaces on a mobile device, comprising the steps of:

loading a software application on the mobile device that requires access to a sensitive application programming interface (API) having a signature identifier;

determining whether the software application includes a digital signature and a signature identification; and

denying the software application access to the sensitive API where the signature identification does not correspond with the signature identifier.

84. (New) The method of claim 83, comprising the additional step of:

purging the software application from the mobile device where the signature identification does not correspond with the signature identifier.

85. (New) The method of claim 83, wherein the digital signature and the signature identification are generated by a code signing authority.

86. (New) The method of claim 83, comprising the additional steps of:

verifying the authenticity of the digital signature where the signature identification corresponds with the signature identifier.; and

denying the software application access to the sensitive API where the digital signature is not authenticated.

87. (New) The method of claim 86, comprising the additional step of:

purging the software application from the mobile device where the digital signature is not authenticated.

88.   (New) The method of claim 86, wherein the digital signature is generated by applying a private signature key to a hash of the software application, and wherein the step of verifying the authenticity of the digital signature is performed by a method comprising the steps of:

storing a public signature key that corresponds to the private signature key on the mobile device;

generating a hash of the software application to obtain a generated hash;

applying the public signature key to the digital signature to obtain a recovered hash; and

comparing the generated hash with the recovered hash.

89.   (New) The method of claim 88, wherein the digital signature is generated by calculating a hash of the software application and applying the private signature key.

90.   (New) The method of claim 83, comprising the additional step of:

displaying a description string that notifies a user of the mobile device that the software application requires access to the sensitive API.

91.   (New) The method of claim 90, comprising the additional step of:

receiving a command from the user granting or denying the software application access to the sensitive API.

92.   (New) A method of controlling access to an application programming interface (API) having a signature identifier on a mobile device by a software application created by a software developer, comprising the steps of:

receiving the software application from the software developer;

determining whether the software application satisfies at least one criterion;

appending a digital signature and a signature identification to the software application where the software application satisfies at least one criterion;;

verifying the authenticity of the digital signature appended to the software application where the signature identification corresponds with the signature identifier; and

providing access to the API to software applications where the digital signature is authenticated.

93.   (New) The method of claim 92, wherein the step of determining whether the software application satisfies at least one criterion is performed by a code signing authority.

94.   (New) The method of claim 92, wherein the step of appending the digital signature and the signature identification to the software application includes generating the digital signature comprising the steps of:

calculating a hash of the software application; and

applying a signature key to the hash of the software application to generate the digital signature.

95.   (New) The method of claim 94, wherein the hash of the software application is calculated using the Secure Hash Algorithm (SHA1).

96.   (New) The method of claim 94, wherein the step of verifying the authenticity of the digital signature comprises the steps of:

providing a corresponding signature key on the mobile device;

calculating the hash of the software application on the mobile device to obtain a calculated hash;

applying the corresponding signature key to the digital signature to obtain a recovered hash; and

authenticating the digital signature by comparing the calculated hash with the recovered hash.

97.   (New) The method of claim 96, comprising the further step of denying the software application access to the API where the digital signature is not authenticated.

98.   (New) The method of claim 96, wherein the signature key is a private signature key and the corresponding signature key is a public signature key.

99.   (New) A method of controlling access to a sensitive application programming interface (API) having a signature identifier on a mobile device, comprising the steps of:

registering one or more software developers that are trusted to develop software applications which access the sensitive API;

receiving a hash of a software application;

determining whether the hash was sent by a registered software developer; and

generating a digital signature using the hash of the software application and a signature identification corresponding to the signature identifier where the hash was sent by the registered software developer;

wherein

the digital signature and the signature identification are appended to the software application; and

the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application where the signature identification corresponds with the signature identifier.

100.   (New) The method of claim 99, wherein the step of generating the digital signature is performed by a code signing authority.

101.   (New) The method of claim 99, wherein the step of generating the digital signature is performed by applying a signature key to the hash of the software application.

102.   (New) The method of claim 101, wherein the mobile device verifies the authenticity of the digital signature by performing the additional steps of:

providing a corresponding signature key on the mobile device;

calculating the hash of the software application on the mobile device to obtain a calculated hash;

applying the corresponding signature key to the digital signature to obtain a recovered hash;

determining whether the digital signature is authentic by comparing the calculated hash with the recovered hash; and

denying the software application access to the sensitive API where the digital signature is not authenticated.

103. (New) A method of restricting access to application programming interfaces on a mobile device, comprising the steps of:

loading a software application having a digital signature and a signature identification on the mobile device that requires access to one or more application programming interfaces (APIs) having at least one signature identifier;

authenticating the digital signature where the signature identification corresponds with the signature identifier; and

denying the software application access to the one or more APIs where the software application does not include an authentic digital signature .

104. (New) The method of claim 103, wherein the digital signature and signature identification are associated with a type of mobile device.

105. (New) The method of claim 103, comprising the additional step of:

purging the software application from the mobile device where the software application does not include an authentic digital signature. .

106. (New) The method of claim 103, wherein:

the software application includes a plurality of digital signatures and signature

identifications; and

the plurality of digital signatures and signature identifications includes digital signatures and signature identifications respectively associated with different types of mobile devices.

107. (New) The method of claim 106, wherein each of the plurality of digital signatures and associated signature identifications are generated by a respective corresponding code signing authority.

108. (New) The method of claim 103, wherein the step of determining whether the software application includes an authentic digital signature comprises the additional steps of:

verifying the authenticity of the digital signature where the signature identification corresponds with respective ones of the at least one signature identifier.

109. (New) The method of claim 107, wherein each of the plurality of digital signatures and signature identifications are generated by its corresponding code signing authority by applying a respective private signature key associated with the code signing authority to a hash of the software application.

110. (New) The method of claim 103, wherein the step of authenticating the digital signature where the signature identification corresponds with the signature identifier comprises the steps of:

verifying that the signature identification corresponds with the signature identifier authenticating the digital signature where signature identification corresponds with the signature identifier comprising the steps of:

storing a public signature key on a mobile device that corresponds to the private signature key associated with the code signing authority which generates the digital signature;

generating a hash of the software application to obtain a generated hash;

applying the public signature key to the digital signature to obtain a recovered hash; and

comparing the generated hash with the recovered hash.

111. (New) The method of claim 103, wherein:

the mobile device includes a plurality of APIs;

at least one of the plurality of APIs is classified as sensitive;

access to any of the plurality of APIs requires an authentic global signature;

access to each of the plurality of sensitive APIs requires an authentic global signature and an authentic digital signature associated with a signature identification;

the step of determining whether the software application includes an authentic digital signature and signature identification comprises the steps of: