



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

Table with 5 columns: APPLICATION NO., ISSUE DATE, PATENT NO., ATTORNEY DOCKET NO., CONFIRMATION NO.
Row 1: 10/381,219, 07/16/2013, 8489868, 13210-1465/KL, 9761

95866 7590 06/26/2013
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 1974 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

David P Yach, Waterloo, ON, CANADA;
Michael S Brown, Waterloo, ON, CANADA;
Herbert A Little, Waterloo, ON, CANADA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No. : 10/381,219
Applicant : David P. YACH et al.
Filed : March 20, 2003
TC/A.U. : 2431
Examiner : Jeremiah L. AVERY
Docket No. : 10289-US-PCT
Customer No. : 95866
Confirmation No. : 9761
For : *SOFTWARE CODE SIGNING SYSTEM AND METHOD*

AMENDMENT AFTER ALLOWANCE

VIA USPTO ELECTRONIC FILE SYSTEM

Mail Stop Amendment

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

ATTENTION: Examiner Jeremiah L. AVERY, Tel. No. 571-272-8627

Sir:

In response to the Notice of Allowance dated March 28, 2013, please enter and consider the following response with amendment and remarks as follows:

Amendment to Claims begins on page 2

Remarks begin on page 26

OK TO ENTER: /J.A./

CERTIFICATE OF TRANSMISSION

In accordance with 37 CFR 1.8, I hereby certify that this correspondence is being 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 or facsimile transmitted or submitted under electronic filing system to the U.S. Patent and Trademark Office on the date: June 3, 2013.

By: Jon A. Gibbons

Signature: / Jon A. Gibbons /
(Applicant, Assignee, or Representative)



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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/381,219 03/20/2003 David P Yach 13210-1465/KL 9761

95866 7590 06/17/2013
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

Table with 1 column: EXAMINER

AVERY, JEREMIAH L

Table with 2 columns: ART UNIT, PAPER NUMBER

2431

Table with 2 columns: NOTIFICATION DATE, DELIVERY MODE

06/17/2013

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):

ptoboca@fggbb.com
portfolioprossecution@blackberry.com

Response to Rule 312 Communication	Application No. 10/381,219	Applicant(s) YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

1. The amendment filed on 03 June 2013 under 37 CFR 1.312 has been considered, and has been:
- a) entered.
 - b) entered as directed to matters of form not affecting the scope of the invention.
 - c) disapproved because the amendment was filed after the payment of the issue fee.
Any amendment filed after the date the issue fee is paid must be accompanied by a petition under 37 CFR 1.313(c)(1) and the required fee to withdraw the application from issue.
 - d) disapproved. See explanation below.
 - e) entered in part. See explanation below.

/NATHAN FLYNN/
Supervisory Patent Examiner, Art Unit 2431

/Jeremiah Avery/
Examiner, Art Unit 2431

5. **Change in Entity Status** (from status indicated above)

- Applicant certifying micro entity status. See 37 CFR 1.29
- Applicant asserting small entity status. See 37 CFR 1.27
- Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see form PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature /Jon A. Gibbons/

Date 6/3/2013

Typed or printed name Jon A. Gibbons

Registration No. 37333

This collection of information is required by 37 CFR 1.311. 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 12 minutes 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

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:	Jon A. Gibbons/KAREN TARAGOWSKI
Attorney Docket Number:	13210-1465/KL

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:				
Utility Appl Issue Fee	1501	1	1780	1780
Publ. Fee- Early, Voluntary, or Normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				2080

Electronic Acknowledgement Receipt

EFS ID:	15935273
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:	95866
Filer:	Jon A. Gibbons/KAREN TARAGOWSKI
Filer Authorized By:	Jon A. Gibbons
Attorney Docket Number:	13210-1465/KL
Receipt Date:	03-JUN-2013
Filing Date:	20-MAR-2003
Time Stamp:	17:13:09
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	\$2080
RAM confirmation Number	4734
Deposit Account	501556
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. 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	Amendment after Notice of Allowance (Rule 312)	10289-US-PCT_312Amendment_6-3-13.pdf	75819 c819923b036ed4ed8673586545353514a301af3b	no	26
Warnings:					
Information:					
2	Issue Fee Payment (PTO-85B)	10289-US-PCT_IssueFeeTransmittal_6-3-13.pdf	106194 8e2f1a5564399c5d960628d06d72006976443ea	no	2
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	32315 49585b5e12237621ffbecf6076495851380a00b4	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			214328		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No. : 10/381,219
Applicant : David P. YACH et al.
Filed : March 20, 2003
TC/A.U. : 2431
Examiner : Jeremiah L. AVERY
Docket No. : 10289-US-PCT
Customer No. : 95866
Confirmation No. : 9761
For : *SOFTWARE CODE SIGNING SYSTEM AND METHOD*

AMENDMENT AFTER ALLOWANCE

VIA USPTO ELECTRONIC FILE SYSTEM

Mail Stop Amendment

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

ATTENTION: Examiner Jeremiah L. AVERY, Tel. No. 571-272-8627

Sir:

In response to the Notice of Allowance dated March 28, 2013, please enter and consider the following response with amendment and remarks as follows:

Amendment to Claims begins on page 2

Remarks begin on page 26

CERTIFICATE OF TRANSMISSION

In accordance with 37 CFR 1.8, I hereby certify that this correspondence is being 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 or facsimile transmitted or submitted under electronic filing system to the U.S. Patent and Trademark Office on the date: June 3, 2013.

By: Jon A. Gibbons

Signature: / Jon A. Gibbons /
(Applicant, Assignee, or Representative)

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-165 (Cancelled without prejudice).

166. (Previously Presented) 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 of the mobile device, the operations comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

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, the operations further comprise: preventing execution of the software application.

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, the operations further comprise: denying the software application access to the sensitive API.

169. (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, the operations further comprise: purging 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 successfully verified, the operations further comprise: preventing execution of the software application.

171. (Previously Presented) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: denying the software application access to the sensitive API.

172. (Previously Presented) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: purging the software application 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 software application to access the sensitive API.

174. (Previously Presented) The mobile device of claim 166, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

175. (Previously Presented) The mobile device of claim 166, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 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 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. (**Currently Amended**) The mobile device of claim 166, wherein 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

190. (Previously Presented) The mobile device of claim 166, wherein the operations further comprise obtaining the public key from a public key repository.

191. (Previously Presented) A system for controlling access to an application platform on a mobile device, 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) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature, the mobile device allowing the software application access to the sensitive API.

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, the operations further comprise: preventing execution of the software application.

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, the operations further comprise: denying the software application access to the sensitive API.

194. (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, the operations further comprise: purging 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 successfully verified, the operations further comprise: preventing execution of the software application.

196. (Previously Presented) The system of claim 191, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: denying the software application access to the sensitive API.

197. (Previously Presented) The system of claim 191, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: purging the software application 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 software application to access the sensitive API.

199. (Previously Presented) The system of claim 191, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

200. (Previously Presented) The system of claim 191, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 the sensitive API, and based upon a determination that the software application should be given access to the sensitive API, the code signing authority accepts the software application and generates the digital signature that is included with the software application.

203. (Previously Presented) The system of claim 191, wherein the operations 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 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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

215. (Previously Presented) The system of claim 191, wherein the operations further comprise obtaining the public key from a public key repository.

216. (Previously Presented) A non-transitory computer-readable storage medium encoded with instructions that when executed on one or more processors of a mobile device, cause the mobile device to perform instructions for controlling access to an application platform of the mobile device, the instructions comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using the public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

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, the instructions further comprising: preventing execution of the software application.

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, the instructions further comprising: denying the software application access to the sensitive API.

219. (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, the instructions further comprising: purging 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 successfully verified, the instructions further comprising: preventing execution of the software application.

221. (Previously Presented) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified, the instructions further comprising: denying the software application access to the sensitive API.

222. (Previously Presented) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified, the instructions further comprising: purging the software application 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 software application to access the sensitive API.

224. (Previously Presented) The computer-readable storage medium of claim 216, wherein at least some of the instructions are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

225. (Previously Presented) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying the private key to a first hash of

the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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, the instructions 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 includes mobile device hardware.

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 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 software application further includes a signature identification, and 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, the instructions further comprising obtaining the public key from a public key repository.

241. (Previously Presented) A method for controlling access to an application platform of a mobile device, comprising:

- storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

- receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

- determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

- the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

- based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

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, the method further comprises: preventing execution of the software application.

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, the method further comprises: denying the software application access to the sensitive API.

244. (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, the method further comprises: purging 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 successfully verified, the method further comprises: preventing execution of the software application.

246. (Previously Presented) The mobile device of claim 241, wherein based upon a determination that the digital signature is not successfully verified, the method further comprises: denying the software application access to the sensitive API.

247. (Previously Presented) The method of claim 241, wherein based upon a determination that the digital signature is not successfully verified, the method further comprises: purging the software application 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 software application to access the sensitive API.

249. (Previously Presented) The method of claim 241, wherein at least some operations of the method are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

250. (Previously Presented) The method of claim 241, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 the sensitive API, wherein based upon a determination that the software application should be given access to the sensitive API, the code signing authority accepts the software application and generates the digital signature that is included with the software application.

253. (Previously Presented) The method of claim 241, further comprising:
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 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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

265. (Previously Presented) The method of claim 241, further comprising obtaining the public key from a public key repository.

266. (Previously Presented) The device of claim 166, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

267. (Previously Presented) The system of claim 191, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

268. (Previously Presented) The computer-readable storage medium of claim 216, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

269. (Previously Presented) The method of claim 241, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

270. (Previously Presented) The device of claim 166, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

271. (Previously Presented) The system of claim 191, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

272. (Previously Presented) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

273. (Previously Presented) The method of claim 241, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

274. (Previously Presented) The device of claim 166, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

275. (Previously Presented) The system of claim 191, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

276. (Previously Presented) The computer-readable storage medium of claim 216, wherein the instructions further comprises: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

277. (Previously Presented) The method of claim 241, further comprising: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

278. (Previously Presented) The mobile device of claim 166, wherein the sensitive API is associated with the public key.

279. (Previously Presented) The mobile device of claim 166, wherein the sensitive API and the public key are included in an API library.

280. (Previously Presented) The mobile device of claim 166, wherein the plurality of APIs comprises at least one non-sensitive API.

281. (Previously Presented) The mobile device of claim 280, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

282. (Previously Presented) The mobile device of claim 280, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

283. (Previously Presented) The mobile device of claim 166, wherein the software application includes a plurality of digital signatures.

284. (Previously Presented) The mobile device of claim 166, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

285. (Previously Presented) The mobile device of claim 166, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:

- using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and
- based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

286. (Previously Presented) The system of claim 191, wherein the sensitive API is associated with the public key.

287. (Previously Presented) The system of claim 191, wherein the sensitive API and the public key are included in an API library.

288. (Previously Presented) The system of claim 191, wherein the plurality of APIs comprises at least one non-sensitive API.

289. (Previously Presented) The system of claim 288, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

290. (Previously Presented) The system of claim 288, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

291. (Previously Presented) The system of claim 191, wherein the software application includes a plurality of digital signatures.

292. (Previously Presented) The system of claim 191, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

293. (Previously Presented) The system of claim 191, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:

- using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and
- based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

294. (Previously Presented) The computer-readable storage medium of claim 216, wherein the sensitive API is associated with the public key.

295. (Previously Presented) The computer-readable storage medium of claim 216, wherein the sensitive API and the public key are included in an API library.

296. (Previously Presented) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises at least one non-sensitive API.

297. (Previously Presented) The computer-readable storage medium of claim 296, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the instructions further comprise: denying the software application access to the sensitive API; and allowing the software application access to the at least one non-sensitive API.

298. (Previously Presented) The computer-readable storage medium of claim 296, wherein based upon a determination that the digital signature is not successfully verified, the instructions further comprise:
denying the software application access to the sensitive API; and
allowing the software application access to the at least one non-sensitive API.

299. (Previously Presented) The computer-readable storage medium of claim 216, wherein the software application includes a plurality of digital signatures.

300. (Previously Presented) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

301. (Previously Presented) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:

using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and

based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

302. (Previously Presented) The method of claim 241, wherein the sensitive API is associated with the public key.

303. (Previously Presented) The method of claim 241, wherein the sensitive API and the public key are included in an API library.

304. (Previously Presented) The method of claim 241, wherein the plurality of APIs comprises at least one non-sensitive API.

305. (Previously Presented) The method of claim 304, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the method further comprising:

denying the software application access to the sensitive API; and

allowing the software application access to the at least one non-sensitive API.

306. (Previously Presented) The method of claim 304, wherein based upon a determination that the digital signature is not successfully verified, the method further comprising:

denying the software application access to the sensitive API; and

allowing the software application access to the at least one non-sensitive API.

307. (Previously Presented) The method of claim 241, wherein the software application includes a plurality of digital signatures.

308. (Previously Presented) The method of claim 241, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

309. (Previously Presented) The method of claim 241, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, the method further comprising:

 using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and

 based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

REMARKS

The Notice of Allowance dated March 28, 2013 has been carefully studied. Applicant would like to thank Examiner Jeremiah L. AVERY for indicating the allowable subject matter of claims 166-309. Claim 188 has been amended to maintain strict antecedent basis. By virtue of this response and amendment claims 166-309 are pending. Reconsideration and allowance of the pending claims in view of the above amendments and the following remarks are respectfully requested.

If the Examiner believes that there are any informalities that can be corrected by Examiner's amendment, or that in any way it would help expedite the prosecution of the patent application, a telephone call to the undersigned at (561) 989-9811 is respectfully solicited.

The Commissioner is hereby authorized to charge any fees that may be required or credit any overpayment to Deposit Account **50-1556** (Attorney Docket No. 10289-US-PCT).

Respectfully submitted,

Date: June 3, 2013

By: /Jon A. Gibbons/
Jon A. Gibbons
(Reg. No. 37,333)
Attorney for Applicant

Fleit Gibbons Gutman
Bongini & Bianco P.L.
One Boca Commerce Center
551 N.W. 77th Street, Suite 111
Boca Raton, Florida 33487
Telephone: (561) 989-9811
Facsimile: (561) 989-9812
www.FGGBB.com



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

NOTICE OF ALLOWANCE AND FEE(S) DUE

95866 7590 03/28/2013
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

EXAMINER

AVERY, JEREMIAH L

ART UNIT PAPER NUMBER

2431

DATE MAILED: 03/28/2013

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/381,219 03/20/2003 David P Yach 13210-1465/KL 9761

TITLE OF INVENTION: SOFTWARE CODE SIGNING SYSTEM AND METHOD

Table with 7 columns: APPLN. TYPE, ENTITY STATUS, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE
nonprovisional UNDISCOUNTED \$1780 \$300 \$0 \$2080 06/28/2013

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

5. **Change in Entity Status** (from status indicated above)

- Applicant certifying micro entity status. See 37 CFR 1.29
- Applicant asserting small entity status. See 37 CFR 1.27
- Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see form PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____

Date _____

Typed or printed name _____

Registration No. _____

This collection of information is required by 37 CFR 1.311. 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 12 minutes 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.

95866 7590 03/28/2013
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

EXAMINER

AVERY, JEREMIAH L

ART UNIT PAPER NUMBER

2431

DATE MAILED: 03/28/2013

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 1626 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 1626 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)	
	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--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to the RCE filed on 03/18/13.
2. An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
3. The allowed claim(s) is/are 166-309. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.
4. 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 the:
 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)).

* Certified copies not received: ____.

Applicant has **THREE MONTHS FROM THE "MAILING DATE"** of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in **ABANDONMENT** of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) 2. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>20130318</u> 3. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material 4. <input type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date ____ . | <ol style="list-style-type: none"> 5. <input type="checkbox"/> Examiner's Amendment/Comment 6. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance 7. <input type="checkbox"/> Other ____. |
|--|---|

/Jeremiah Avery/
Examiner, Art Unit 2431

/NATHAN FLYNN/
Supervisory Patent Examiner, Art Unit 2431

Priority

1. This Application, 10/381219, is a national stage entry of PCT/CA01/01344, International Filing Date: 09/20/2001.
2. PCT/CA01/01344 claims priority from Provisional Application 60234152, filed 09/21/2000.

Continued Examination Under 37 CFR 1.114

3. 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 allowance or after an Office action under *Ex Parte Quayle*, 25 USPQ 74, 453 O.G. 213 (Comm'r Pat. 1935). 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, prosecution in this application has been reopened pursuant to 37 CFR 1.114. Applicant's submission filed on 03/18/13 has been entered.

Drawings

4. The drawings were received on 03/18/13. These drawings are accepted.

Examiner's Statement of Reasons for Allowance

5. Dependent claims 278-309 have been added.
6. Claims 166-309 are allowed over the prior art.
7. This action is in reply to the applicant's correspondence on 03/18/13.
8. The following is an examiner's statement of reasons for the indication of allowable claimed subject matter.

9. As per independent claims 166, 191, 216 and 241, generally, the prior art of record, United States Patent No. 6,795,919 to Gibbs et al., and United States Patent No. 6,587,837 to Spagna et al., fails to teach alone, or in combination, other than via hindsight, at the time of the invention, the features as discussed and remarked upon in the response of 03/18/13.

10. The Applicant's amendments to the claims and presented arguments distinguish the claimed invention over the prior art and place this application in condition for allowance.

Conclusion

11. 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.

12. 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.

13. 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

Application/Control Number: 10/381,219

Page 4

Art Unit: 2431

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 2431

Notice of References Cited	Application/Control No. 10/381,219	Applicant(s)/Patent Under Reexamination YACH ET AL.	
	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-6,574,609	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-6,233,683	05-2001	Chan et al.	713/172
*	G	US-6,390,374	05-2002	Carper et al.	235/492
*	H	US-6,374,357	04-2002	Mohammed et al.	726/5
*	I	US-6,345,256	02-2002	Milsted et al.	705/64
*	J	US-6,697,948	02-2004	Rabin et al.	726/30
*	K	US-6,212,636	04-2001	Boyle et al.	713/168
*	L	US-6,748,541	06-2004	Margalit et al.	726/9
	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	Maass, Henning. Open Mobility Management Platform with Directory-Based Architecture and Signalling Protocols. 1998 IEEE Open Architectures and Network Programming. Pub Date: 1998. Relevant Pages: 72-87. Found on the World Wide Web at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=662045
	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.

<i>Index of Claims</i> 	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	03/20/2013									
	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	03/20/2013									
	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. 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	03/20/2013									
	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. 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	03/20/2013									
	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. 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


<input type="checkbox"/> Claims renumbered in the same order as presented by applicant			<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47	
CLAIM		DATE						
Final	Original	03/20/2013						
	145	-						
	146	-						
	147	-						
	148	-						
	149	-						
	150	-						
	151	-						
	152	-						
	153	-						
	154	-						
	155	-						
	156	-						
	157	-						
	158	-						
	159	-						
	160	-						
	161	-						
	162	-						
	163	-						
	164	-						
	165	-						
1	166	=						
2	167	=						
3	168	=						
4	169	=						
5	170	=						
6	171	=						
7	172	=						
8	173	=						
9	174	=						
10	175	=						
11	176	=						
12	177	=						
13	178	=						
14	179	=						
15	180	=						

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	03/20/2013									
16	181	=									
17	182	=									
18	183	=									
19	184	=									
20	185	=									
21	186	=									
22	187	=									
23	188	=									
24	189	=									
25	190	=									
26	191	=									
27	192	=									
28	193	=									
29	194	=									
30	195	=									
31	196	=									
32	197	=									
33	198	=									
34	199	=									
35	200	=									
36	201	=									
37	202	=									
38	203	=									
39	204	=									
40	205	=									
41	206	=									
42	207	=									
43	208	=									
44	209	=									
45	210	=									
46	211	=									
47	212	=									
48	213	=									
49	214	=									
50	215	=									
51	216	=									

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	03/20/2013									
52	217	=									
53	218	=									
54	219	=									
55	220	=									
56	221	=									
57	222	=									
58	223	=									
59	224	=									
60	225	=									
61	226	=									
62	227	=									
63	228	=									
64	229	=									
65	230	=									
66	231	=									
67	232	=									
68	233	=									
69	234	=									
70	235	=									
71	236	=									
72	237	=									
73	238	=									
74	239	=									
75	240	=									
76	241	=									
77	242	=									
78	243	=									
79	244	=									
80	245	=									
81	246	=									
82	247	=									
83	248	=									
84	249	=									
85	250	=									
86	251	=									
87	252	=									

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	03/20/2013									
88	253	=									
89	254	=									
90	255	=									
91	256	=									
92	257	=									
93	258	=									
94	259	=									
95	260	=									
96	261	=									
97	262	=									
98	263	=									
99	264	=									
100	265	=									
101	266	=									
102	267	=									
103	268	=									
104	269	=									
105	270	=									
106	271	=									
107	272	=									
108	273	=									
109	274	=									
110	275	=									
111	276	=									
112	277	=									
113	278	=									
114	279	=									
115	280	=									
116	281	=									
117	282	=									
118	283	=									
119	284	=									
120	285	=									
121	286	=									
122	287	=									
123	288	=									

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	03/20/2013									
124	289	=									
125	290	=									
126	291	=									
127	292	=									
128	293	=									
129	294	=									
130	295	=									
131	296	=									
132	297	=									
133	298	=									
134	299	=									
135	300	=									
136	301	=									
137	302	=									
138	303	=									
139	304	=									
140	305	=									
141	306	=									
142	307	=									
143	308	=									
144	309	=									

Search Notes 	Application/Control No. 10381219	Applicant(s)/Patent Under Reexamination YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner
none	none	3/20/2013	JLA

SEARCH NOTES		
Search Notes	Date	Examiner
Updated EAST Search	3/20/2013	JLA
Updated Keyword Search within Class 711, subclass 100, Class 713, subclasses 1, 176, 187 and 189Class 395, subclass 682 and Class 719, subclass 328	3/20/2013	JLA
Updated Inventor Search	3/20/2013	JLA
Assignee Search	3/20/2013	JLA
IEEE Search	3/20/2013	JLA

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

--	--

INTERFERENCE SEARCH

US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
none	((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) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))))).clm.	3/20/2013	JLA

--	--

Receipt date: 03/18/2013

10381219 - GALL:2431

Doc code: IDS

Doc description: Information Disclosure Statement (IDS) Filed

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.

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	10289-US-PCT		

U.S.PATENTS							Remove	
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	6223291		2001-04-24	Puhl et al.			
	2	6289382		2001-09-11	Bowman-Amuah			
	3	6526513		2003-02-25	Shrader et al.			
	4	6697948		2004-02-24	Rabin et al.			
	5	6766353		2004-07-20	Lin et al.			
If you wish to add additional U.S. Patent citation information please click the Add button.							Add	
U.S.PATENT APPLICATION PUBLICATIONS							Remove	
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.							Add	
FOREIGN PATENT DOCUMENTS							Remove	

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	10289-US-PCT		

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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							<input type="checkbox"/>

If you wish to add additional Foreign Patent Document citation information please click the Add button **Add**

NON-PATENT LITERATURE DOCUMENTS Remove

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	Office Action dated May 11, 2012 for U.S. Patent Application Serial No. 13/413,173.	<input type="checkbox"/>
	2	Office Action dated November 30, 2012 for U.S. Patent Application Serial No. 13/413,173.	<input type="checkbox"/>
	3	Java Platform Standard Ed. 6, http://docs.oracle.com/javase/6/docs/api/java/lang/reflect/Method.html (last visited Nov. 3, 2012).	<input type="checkbox"/>
	4	Application programming interface, http://en.wikipedia.org/w/index.php?title=Application_programming_interface&oldid=520968418 (last visited Nov. 3, 2012).	<input type="checkbox"/>
	5	ETSI TS 123 057 v3.3.0 (2000-10-16).	<input type="checkbox"/>
	6	DEVANBU, P.T., et al., "Techniques for trusted software engineering." Proceedings of the 20th International Conference on Software Engineering, p. 126-135. April 19-25, Kyoto, Japan.	<input type="checkbox"/>
	7	ETSI TS 123 057 v3.2.0 (2000-6-23).	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

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	10289-US-PCT	

EXAMINER SIGNATURE			
Examiner Signature	/Jeremiah Avery/	Date Considered	03/20/2013
<p>*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.</p>			
<p><small>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.</small></p>			

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	10289-US-PCT	

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Jon A. Gibbons/	Date (YYYY-MM-DD)	2013-03-18
Name/Print	Jon A. Gibbons	Registration Number	37333

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

SEARCH RESULTS

You searched for: (((API or (Application near programming near interface))) AND access) AND public and private) AND key)

You Refined by:

Publisher: IEEE (x)

Content Type: Conference Publications (x), Journals & Magazines (x)

Publication Year: 1872 - 2000 (x)

EAST Search History

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	3	((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) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))))).clm.	US-PGPUB; USPAT; UPAD	OR	ON	2013/03/20 15:43

3/ 20/ 2013 3:57:54 PM

C:\ Users\ javery\ Documents\ EAST\ Workspaces\ 10381219.wsp

EAST Search History


EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	1243	((@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	2013/03/20 15:44
L3	121	(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 L2	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L4	89	L3 and ((authentic\$ or verify\$ or verification) with signature)	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L5	108615	((@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	2013/03/20 15:44
L6	1171	L5 and ((unauthentic\$ or unverify\$ or unverifi\$) or ("not" near (authentic\$ or verify\$ or verification)))	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L7	609	L6 and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L8	606	L7 and access\$	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L9	164	L8 and (API or (application near program\$4 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L10	3	L9 and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping)))	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:44
L11	2	(YACH-DAVID-P.in. or BROWN-MICHAEL-S.in. or LITTLE-HERBERT-A.in.) and ((API or (Application near programming near	US-PGPUB; USPAT;	OR	ON	2013/03/20 15:56

		interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))).clm.	EPO			
L12	4	(research-in-motion-limited.as.) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))).clm.	US-PGPUB; USPAT; EPO	OR	ON	2013/03/20 15:57


3/ 20/ 2013 3:57:44 PM

C:\ Users\ javery\ Documents\ EAST\ Workspaces\ 10381219.wsp

Issue Classification 	Application/Control No. 10381219	Applicant(s)/Patent Under Reexamination YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant		<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47									
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		41		81		121		161	36	201	76	241	116	281
	2		42		82		122		162	37	202	77	242	117	282
	3		43		83		123		163	38	203	78	243	118	283
	4		44		84		124		164	39	204	79	244	119	284
	5		45		85		125		165	40	205	80	245	120	285
	6		46		86		126	1	166	41	206	81	246	121	286
	7		47		87		127	2	167	42	207	82	247	122	287
	8		48		88		128	3	168	43	208	83	248	123	288
	9		49		89		129	4	169	44	209	84	249	124	289
	10		50		90		130	5	170	45	210	85	250	125	290
	11		51		91		131	6	171	46	211	86	251	126	291
	12		52		92		132	7	172	47	212	87	252	127	292
	13		53		93		133	8	173	48	213	88	253	128	293
	14		54		94		134	9	174	49	214	89	254	129	294
	15		55		95		135	10	175	50	215	90	255	130	295
	16		56		96		136	11	176	51	216	91	256	131	296
	17		57		97		137	12	177	52	217	92	257	132	297
	18		58		98		138	13	178	53	218	93	258	133	298
	19		59		99		139	14	179	54	219	94	259	134	299
	20		60		100		140	15	180	55	220	95	260	135	300
	21		61		101		141	16	181	56	221	96	261	136	301
	22		62		102		142	17	182	57	222	97	262	137	302
	23		63		103		143	18	183	58	223	98	263	138	303
	24		64		104		144	19	184	59	224	99	264	139	304
	25		65		105		145	20	185	60	225	100	265	140	305
	26		66		106		146	21	186	61	226	101	266	141	306
	27		67		107		147	22	187	62	227	102	267	142	307
	28		68		108		148	23	188	63	228	103	268	143	308
	29		69		109		149	24	189	64	229	104	269	144	309
	30		70		110		150	25	190	65	230	105	270		
	31		71		111		151	26	191	66	231	106	271		
	32		72		112		152	27	192	67	232	107	272		
	33		73		113		153	28	193	68	233	108	273		
	34		74		114		154	29	194	69	234	109	274		
	35		75		115		155	30	195	70	235	110	275		
	36		76		116		156	31	196	71	236	111	276		
	37		77		117		157	32	197	72	237	112	277		

/JEREMIAH AVERY/ Examiner.Art Unit 2431 (Assistant Examiner)	03/20/13 (Date)	Total Claims Allowed: 144	
/NATHAN FLYNN/ Supervisory Patent Examiner.Art Unit 2431 (Primary Examiner)	03/22/2013 (Date)	O.G. Print Claim(s) 166	O.G. Print Figure 3A

Issue Classification 	Application/Control No. 10381219	Applicant(s)/Patent Under Reexamination YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant		<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47					
38	78	118	158	33	198	73	238	113	278		
39	79	119	159	34	199	74	239	114	279		
40	80	120	160	35	200	75	240	115	280		

/JEREMIAH AVERY/ Examiner.Art Unit 2431 (Assistant Examiner)	03/20/13 (Date)	Total Claims Allowed: 144	
/NATHAN FLYNN/ Supervisory Patent Examiner.Art Unit 2431 (Primary Examiner)	03/22/2013 (Date)	O.G. Print Claim(s) 166	O.G. Print Figure 3A

**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)	10289-US-PCT	Art Unit	2431
First Named Inventor	David P. YACH			Examiner Name	Jeremiah L. 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 _____
- Enclosed
- Amendment/Reply
- Information Disclosure Statement (IDS)
- Affidavit(s)/ Declaration(s)
- Other _____

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 501556

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

- Patent Practitioner Signature
- Applicant Signature

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.

Signature of Registered U.S. Patent Practitioner			
Signature	/Jon A. Gibbons/	Date (YYYY-MM-DD)	2013-03-18
Name	Jon A. Gibbons	Registration Number	37333

This collection of information is required by 37 CFR 1.114. 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 1.11 and 1.14. This collection is estimated to take 12 minutes 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No. : 10/381,219
Applicant : David P. YACH et al.
Filed : March 20, 2003
TC/A.U. : 2431
Examiner : Jeremiah L. AVERY
Docket No. : 10289-US-PCT
Customer No. : 95866
Confirmation No. : 9761
For : *SOFTWARE CODE SIGNING SYSTEM AND METHOD*

RESPONSE WITH AMENDMENT

VIA USPTO ELECTRONIC FILE SYSTEM

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
ATTENTION: EXAMINER Jeremiah L. Avery

Sir:

In response to the Notice of Allowance dated December 19, 2012, please enter and consider the following response with amendment and remarks and information disclosure statement as follows:

Amendment to the Drawings begins on page 2
Amendments to the Specification begins on page 3
Amendment to Claims begins on page 4
Remarks begin on page 28
Replacement Sheet for FIG. 1 attached

CERTIFICATE OF TRANSMISSION

In accordance with 37 CFR 1.8, I hereby certify that this correspondence is being 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 or facsimile transmitted or submitted under electronic filing system to the U.S. Patent and Trademark Office on the date: March 18, 2013.

By: Jon Gibbons

Signature: /Jon Gibbons/
(Applicant, Assignee, or Representative)

Appl. No. «SERIAL»
Docket No. «C_REFERENCE»
Reply to Notice of Allowance December 19, 2012

IN THE DRAWINGS

Applicants respectfully request the Examiner's permission to remove the "12" on the right of the bottom box "device". No new matter has been added.

IN THE SPECIFICATION

Please amend paragraph [0037] U.S. Patent Publication No. 20040025022 as follows:

[0037] 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 software application 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.

No new matter has been added.

Please amend paragraph [0039] U.S. Patent Publication No. 20040025022 as follows:

[0039] 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 with any sensitive APIs to which access is required, in steps 106-116.

No new matter has been added.

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-165 (Cancelled without prejudice).

166. (Previously Presented) 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 of the mobile device, the operations comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

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, the operations further comprise: preventing execution of the software application.

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, the operations further comprise: denying the software application access to the sensitive API.

169. (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, the operations further comprise: purging 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 successfully verified, the operations further comprise: preventing execution of the software application.

171. (Previously Presented) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: denying the software application access to the sensitive API.

172. (Previously Presented) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: purging the software application 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 software application to access the sensitive API.

174. (Previously Presented) The mobile device of claim 166, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

175. (Previously Presented) The mobile device of claim 166, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 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 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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

190. (Previously Presented) The mobile device of claim 166, wherein the operations further comprise obtaining the public key from a public key repository.

191. (Previously Presented) A system for controlling access to an application platform on a mobile device, 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) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature, the mobile device allowing the software application access to the sensitive API.

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, the operations further comprise: preventing execution of the software application.

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, the operations further comprise: denying the software application access to the sensitive API.

194. (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, the operations further comprise: purging 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 successfully verified, the operations further comprise: preventing execution of the software application.

196. (Previously Presented) The system of claim 191, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: denying the software application access to the sensitive API.

197. (Previously Presented) The system of claim 191, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise: purging the software application 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 software application to access the sensitive API.

199. (Previously Presented) The system of claim 191, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

200. (Previously Presented) The system of claim 191, wherein the digital signature is generated by applying the private key to a first hash of the software application; and

the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 the sensitive API, and based upon a determination that the software application should be given access to the sensitive API, the code signing authority accepts the software application and generates the digital signature that is included with the software application.

203. (Previously Presented) The system of claim 191, wherein the operations 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 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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

215. (Previously Presented) The system of claim 191, wherein the operations further comprise obtaining the public key from a public key repository.

216. (**Currently Amended**) A non-transitory computer-readable storage medium encoded with instructions that when executed on one or more processors of a mobile device, cause the mobile device to perform ~~a method~~ instructions for controlling access to an application platform of the mobile device, the ~~method~~ instructions comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using the public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

217. (**Currently Amended**) 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, the ~~method~~ instructions further ~~comprises~~ comprising: preventing execution of the software application.

218. (**Currently Amended**) 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, the ~~method~~ instructions further ~~comprises~~ comprising: denying the software application access to the sensitive API.

219. **(Currently Amended)** 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, the ~~method~~ instructions further ~~comprises~~ comprising: purging the software application from the mobile device.

220. **(Currently Amended)** The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified, the ~~method~~ instructions further ~~comprises~~ comprising: preventing execution of the software application.

221. **(Currently Amended)** The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified, the ~~method~~ instructions further ~~comprises~~ comprising: denying the software application access to the sensitive API.

222. **(Currently Amended)** The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified, the ~~method~~ instructions further ~~comprises~~ comprising: purging the software application 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 software application to access the sensitive API.

224. **(Currently Amended)** The computer-readable storage medium of claim 216, wherein at least some of the ~~operations~~ instructions are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

225. (Previously Presented) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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. (**Currently Amended**) The computer-readable storage medium of claim 216, the instructions 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 includes mobile device hardware.

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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

240. (**Currently Amended**) The computer-readable storage medium of claim 216, ~~wherein the method instructions further comprises comprising obtaining the public key from a public key repository.~~

241. (Previously Presented) A method for controlling access to an application platform of a mobile device, comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, wherein at least one API comprises a sensitive API to which access is restricted;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to the sensitive API stored at the mobile device;

determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device;

the mobile device using a public key of the private key-public key pair to verify the digital signature of the software application; and

based upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to the sensitive API.

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, the method further comprises: preventing execution of the software application.

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, the method further comprises: denying the software application access to the sensitive API.

244. (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, the method further comprises: purging 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 successfully verified, the method further comprises: preventing execution of the software application.

246. (Previously Presented) The mobile device of claim 241, wherein based upon a determination that the digital signature is not successfully verified, the method further comprises: denying the software application access to the sensitive API.

247. (Previously Presented) The method of claim 241, wherein based upon a determination that the digital signature is not successfully verified, the method further comprises: purging the software application 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 software application to access the sensitive API.

249. (Previously Presented) The method of claim 241, wherein at least some operations of the method are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

250. (Previously Presented) The method of claim 241, wherein the digital signature is generated by applying the private key to a first hash of the software application; and the digital signature is verified 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 the private key to a first abridged version of the software application; and the digital signature is verified 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 the sensitive API, wherein based upon a determination that the software application should be given access to the sensitive API, the code signing authority accepts the software application and generates the digital signature that is included with the software application.

253. (Previously Presented) The method of claim 241, further comprising:
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 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 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 software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

265. (Previously Presented) The method of claim 241, further comprising obtaining the public key from a public key repository.

266. (Previously Presented) The device of claim 166, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

267. (Previously Presented) The system of claim 191, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

268. (Previously Presented) The computer-readable storage medium of claim 216, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

269. (Previously Presented) The method of claim 241, wherein verifying the digital signature comprises:

- hashing the software application to obtain a generated hash;
- applying the public key to the digital signature to obtain a recovered hash; and
- comparing the generated hash and the recovered hash.

270. (Previously Presented) The device of claim 166, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

271. (Previously Presented) The system of claim 191, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

272. (Previously Presented) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

273. (Previously Presented) The method of claim 241, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

274. (Previously Presented) The device of claim 166, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

275. (Previously Presented) The system of claim 191, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

276. (**Currently Amended**) The computer-readable storage medium of claim 216, wherein the instructions further comprises: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

277. (Previously Presented) The method of claim 241, further comprising: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

278. (**New**) The mobile device of claim 166, wherein the sensitive API is associated with the public key.

279. (**New**) The mobile device of claim 166, wherein the sensitive API and the public key are included in an API library.

280. (**New**) The mobile device of claim 166, wherein the plurality of APIs comprises at least one non-sensitive API.

281. (**New**) The mobile device of claim 280, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

282. (**New**) The mobile device of claim 280, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

283. **(New)** The mobile device of claim 166, wherein the software application includes a plurality of digital signatures.

284. **(New)** The mobile device of claim 166, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

285. **(New)** The mobile device of claim 166, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:

using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and

based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

286. **(New)** The system of claim 191, wherein the sensitive API is associated with the public key.

287. **(New)** The system of claim 191, wherein the sensitive API and the public key are included in an API library.

288. **(New)** The system of claim 191, wherein the plurality of APIs comprises at least one non-sensitive API.

289. **(New)** The system of claim 288, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the operations further comprise:

denying the software application access to the sensitive API; and

allowing the software application access to the at least one non-sensitive API.

290. **(New)** The system of claim 288, wherein based upon a determination that the digital signature is not successfully verified, the operations further comprise:
denying the software application access to the sensitive API; and
allowing the software application access to the at least one non-sensitive API.

291. **(New)** The system of claim 191, wherein the software application includes a plurality of digital signatures.

292. **(New)** The system of claim 191, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

293. **(New)** The system of claim 191, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:
using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and
based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

294. **(New)** The computer-readable storage medium of claim 216, wherein the sensitive API is associated with the public key.

295. **(New)** The computer-readable storage medium of claim 216, wherein the sensitive API and the public key are included in an API library.

296. **(New)** The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises at least one non-sensitive API.

297. **(New)** The computer-readable storage medium of claim 296, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the instructions further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

298. **(New)** The computer-readable storage medium of claim 296, wherein based upon a determination that the digital signature is not successfully verified, the instructions further comprise:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

299. **(New)** The computer-readable storage medium of claim 216, wherein the software application includes a plurality of digital signatures.

300. **(New)** The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

301. **(New)** The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, wherein the operations further comprise:

- using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and

- based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

302. **(New)** The method of claim 241, wherein the sensitive API is associated with the public key.

303. **(New)** The method of claim 241, wherein the sensitive API and the public key are included in an API library.

304. **(New)** The method of claim 241, wherein the plurality of APIs comprises at least one non-sensitive API.

305. **(New)** The method of claim 304, wherein based upon a determination that the software application requesting access to the sensitive API does not include a signature, the method further comprising:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

306. **(New)** The method of claim 304, wherein based upon a determination that the digital signature is not successfully verified, the method further comprising:

- denying the software application access to the sensitive API; and
- allowing the software application access to the at least one non-sensitive API.

307. **(New)** The method of claim 241, wherein the software application includes a plurality of digital signatures.

308. **(New)** The method of claim 241, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein one or more of the sensitive APIs is associated with a unique digital signature.

309. **(New)** The method of claim 241, wherein the plurality of APIs comprises at least a second sensitive API, wherein the software application includes at least a second digital signature, the method further comprising:

 using a second public key of a second private key-public key pair to verify the second digital signature of the software application; and

 based upon verifying the second digital signature at the mobile device, allowing the software application access to at least the second sensitive API.

REMARKS

The Notice of Allowance dated December 19, 2012 has been studied. Applicant wishes to thank examiner Jeremiah Avery for indicating that claims 166-277 are allowed. Claims 278-309 are newly added dependent claims. Claims 216-222, 224, and 240 are amended to clarify that instructions are being executed from the computer-readable storage medium. By virtue of this Response and Amendment claims 166-309 are pending. An information disclosure statement with a Request for Continued Examination (RCE) is being filed herewith. Continued examination and allowance of the pending claims are respectfully requested.

Support for newly added dependent claims 278, 286, 294, 302 and 279, 287, 295, 303 is found at least in U.S. Patent Publication No. 20040025022 at least FIG. 3A items 20 and 78, FIG. 4 step 106, and paragraphs [0028], [0036], [0037], [0040], [0041]. No new matter has been added.

Support for newly added dependent claims 280, 288, 296, 304 is found at least in U.S. Patent Publication No. 20040025022 at least in paragraphs [0035] and [0036]. No new matter has been added.

Support for newly added dependent claims 281, 289, 297, 305 is found at least in U.S. Patent Publication No. 20040025022 at least FIG. 4 step 102 and paragraph [0039]. No new matter has been added.

Support for newly added dependent claims 282, 290, 298, 306 is found at least in U.S. Patent Publication No. 20040025022 at least in paragraph [0037]. No new matter has been added.

Support for newly added dependent claims 283, 291, 299, 307 is found at least in U.S. Patent Publication No. 20040025022 at least at FIG. 3 and FIG. 3A and in paragraphs [0035] and [0038]. No new matter has been added.

Appl. No. «SERIAL»
Docket No. «C_REFERENCE»
Reply to Notice of Allowance December 19, 2012

Support for newly added dependent claims 284, 292, 300, 308 is found at least in U.S. Patent Publication No. 20040025022 at least at FIG. 3 and FIG. 3A and in paragraph [0035]. No new matter has been added.

Support for newly added dependent claims 285, 293, 301, 309 is found at least in U.S. Patent Publication No. 20040025022 at least at FIG. 3 and FIG. 3A and in paragraph [0035]. No new matter has been added.

If the Examiner believes that there are any informalities that can be corrected by Examiner's amendment, or that in any way it would help expedite the prosecution of the patent application, a telephone call to the undersigned at (561) 989-9811 is respectfully solicited.

The Commissioner is hereby authorized to charge any fees that may be required or credit any overpayment to Deposit Account **50-1556** (Attorney Docket No. 10289-US-PCT).

Respectfully submitted,

Date: March 18, 2013

By: /Jon Gibbons/
Jon A. Gibbons
(Reg. No.37,333)
Attorney for Applicant

Fleit Gibbons Gutman
Bongini & Bianco P.L.
One Boca Commerce Center
551 N.W. 77th Street, Suite 111
Boca Raton, Florida 33487
Telephone: (561) 989-9811
Facsimile: (561) 989-9812

REPLACEMENT SHEET

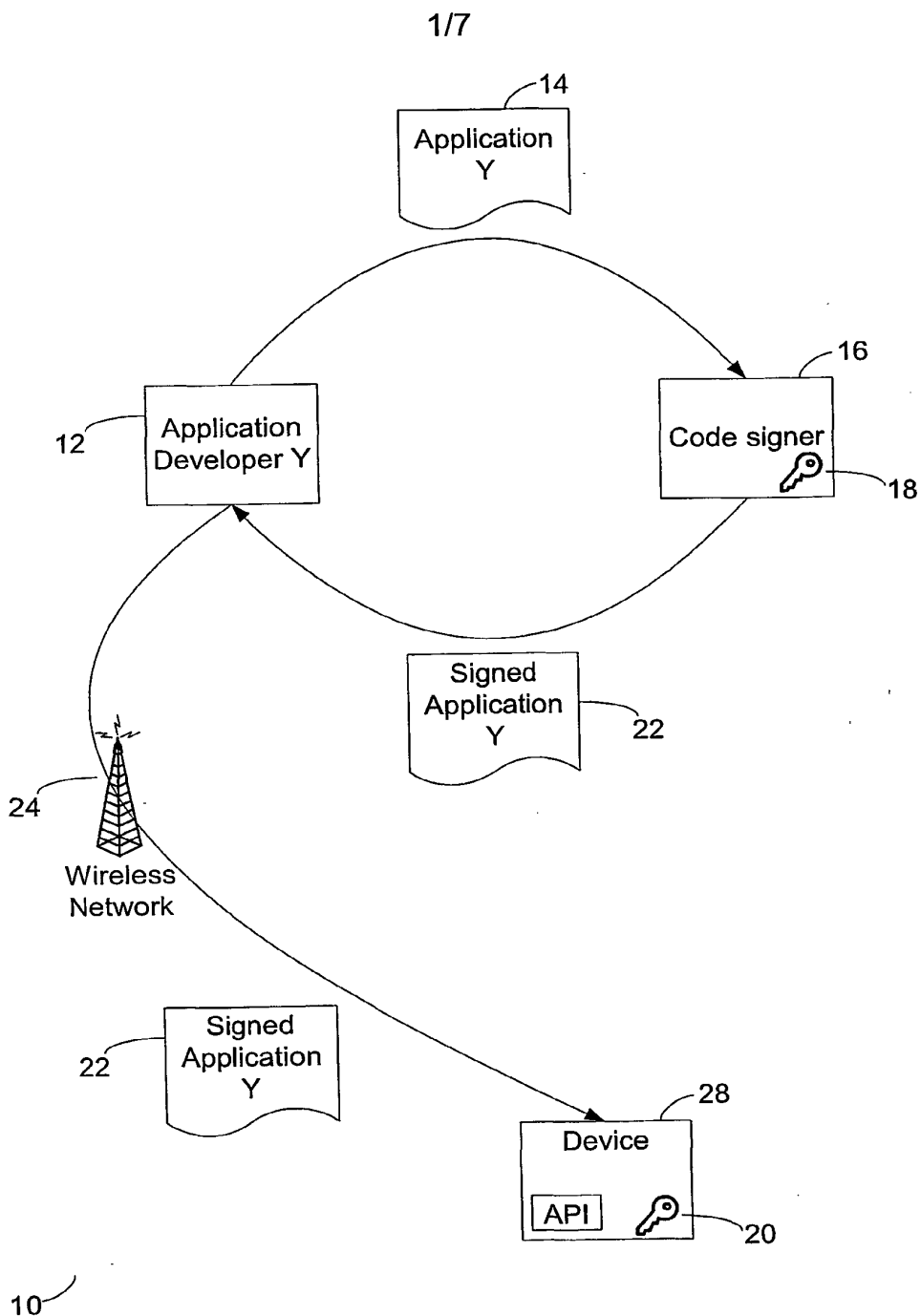


Figure 1

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	10289-US-PCT		

U.S.PATENTS							Remove	
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	6223291		2001-04-24	Puhl et al.			
	2	6289382		2001-09-11	Bowman-Amuah			
	3	6526513		2003-02-25	Shrader et al.			
	4	6697948		2004-02-24	Rabin et al.			
	5	6766353		2004-07-20	Lin et al.			
If you wish to add additional U.S. Patent citation information please click the Add button.							Add	
U.S.PATENT APPLICATION PUBLICATIONS							Remove	
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.							Add	
FOREIGN PATENT DOCUMENTS							Remove	

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	10289-US-PCT		

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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							<input type="checkbox"/>

If you wish to add additional Foreign Patent Document citation information please click the Add button **Add**

NON-PATENT LITERATURE DOCUMENTS Remove

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	Office Action dated May 11, 2012 for U.S. Patent Application Serial No. 13/413,173.	<input type="checkbox"/>
	2	Office Action dated November 30, 2012 for U.S. Patent Application Serial No. 13/413,173.	<input type="checkbox"/>
	3	Java Platform Standard Ed. 6, http://docs.oracle.com/javase/6/docs/api/java/lang/reflect/Method.html (last visited Nov. 3, 2012).	<input type="checkbox"/>
	4	Application programming interface, http://en.wikipedia.org/w/index.php?title=Application_programming_interface&oldid=520968418 (last visited Nov. 3, 2012).	<input type="checkbox"/>
	5	ETSI TS 123 057 v3.3.0 (2000-10-16).	<input type="checkbox"/>
	6	DEVANBU, P.T., et al., "Techniques for trusted software engineering." Proceedings of the 20th International Conference on Software Engineering, p. 126-135. April 19-25, Kyoto, Japan.	<input type="checkbox"/>
	7	ETSI TS 123 057 v3.2.0 (2000-6-23).	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

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	10289-US-PCT

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	10289-US-PCT

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Jon A. Gibbons/	Date (YYYY-MM-DD)	2013-03-18
Name/Print	Jon A. Gibbons	Registration Number	37333

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

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:	Thomas Grzesik/Jon Gibbons			
Attorney Docket Number:	13210-1465/KL			
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	32	62	1984
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:				
Request for Continued Examination	1801	1	930	930
Total in USD (\$)				2914

Electronic Acknowledgement Receipt

EFS ID:	15293057
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:	95866
Filer:	Thomas Grzesik/Jon Gibbons
Filer Authorized By:	Thomas Grzesik
Attorney Docket Number:	13210-1465/KL
Receipt Date:	18-MAR-2013
Filing Date:	20-MAR-2003
Time Stamp:	23:01:32
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	\$2914
RAM confirmation Number	13981
Deposit Account	501556
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
-----------------	----------------------	-----------	----------------------------------	------------------	------------------

1	Request for Continued Examination (RCE)	10289-US-PCT_RCE_3-18-13.pdf	697765 84a08f762d099c1a3eb66685780d5a0d793083d8	no	3
Warnings:					
Information:					
2	Amendment Submitted/Entered with Filing of CPA/RCE	10289-US-PCT_AmendmentMarch2013.pdf	112443 b83e61848c6fbdcaa21b331e537c587c4507ead3	no	29
Warnings:					
Information:					
3	Drawings-only black and white line drawings	ReplacementSheetFIG1.pdf	59893 f07c96a38fae1f8519fda8ac5f9200a87ec679b0	no	1
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Form (SB08)	10289-US-PCT_IDS_3-18-13.pdf	612960 3dbfb0f2d0d356b27750c4f3ceaf772dc0e2ba33	no	5
Warnings:					
Information:					
5	Non Patent Literature	NPL1.pdf	4098352 7b7d86947c5b9035e6d42076f114057639f10333	no	52
Warnings:					
Information:					
6	Non Patent Literature	NPL2.pdf	2542544 db4307215e84199db79206ff6db1328370c296b7	no	33
Warnings:					
Information:					
7	Non Patent Literature	NPL3.pdf	1246675 ad6c3d5654cd66159d7b7ce6c7c7b60991b1080e	no	12
Warnings:					
Information:					
8	Non Patent Literature	NPL4.pdf	188843 337be4b60e0f9c4d2e30ac59f4396c10cbe5c11b	no	2
Warnings:					
Information:					
9	Non Patent Literature	NPL5.pdf	6505984 3beb9779e7ed14039be557a7e4fd21e3ad73b05	no	60
Warnings:					
Information:					

10	Non Patent Literature	NPL6.pdf	2330164 ea58392e0250e58c8add0a3bee9dfc6af2eb345	no	10
Warnings:					
Information:					
11	Non Patent Literature	NPL7.pdf	25485522 53a8324f16e04bea728c5efcf159e359e2025ac2	no	61
Warnings:					
Information:					
12	Fee Worksheet (SB06)	fee-info.pdf	32468 010efb896fba9aac23a6b01f76d576ffa9d1264b	no	2
Warnings:					
Information:					
Total Files Size (in bytes):				43913613	
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875				Application or Docket Number 10/381,219		Filing Date 03/20/2003		<input type="checkbox"/> To be Mailed				
APPLICATION AS FILED – PART I						OTHER THAN						
(Column 1)		(Column 2)		SMALL ENTITY <input type="checkbox"/>		OR		SMALL ENTITY				
FOR	NUMBER FILED	NUMBER EXTRA	RATE (\$)	FEE (\$)	OR	RATE (\$)	FEE (\$)					
<input type="checkbox"/> BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>	N/A	N/A	N/A			N/A						
<input type="checkbox"/> SEARCH FEE <small>(37 CFR 1.16(k), (j), or (m))</small>	N/A	N/A	N/A			N/A						
<input type="checkbox"/> EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>	N/A	N/A	N/A			N/A						
TOTAL CLAIMS <small>(37 CFR 1.16(i))</small>	minus 20 =	*	X \$ =		OR	X \$ =						
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>	minus 3 =	*	X \$ =			X \$ =						
<input type="checkbox"/> APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>	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).											
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT <small>(37 CFR 1.16(j))</small>												
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL						
APPLICATION AS AMENDED – PART II						OTHER THAN						
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR		SMALL ENTITY		
AMENDMENT	03/18/2013	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR	RATE (\$)	ADDITIONAL FEE (\$)		
	Total (37 CFR 1.16(i))	* 144	Minus	** 112	= 32	X \$ =		OR	X \$62=	1984		
	Independent (37 CFR 1.16(h))	* 4	Minus	***4	= 0	X \$ =		OR	X \$250=	0		
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))								OR			
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								OR			
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	1984		
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR	RATE (\$)	ADDITIONAL FEE (\$)		
	Total (37 CFR 1.16(i))	*	Minus	**	=	X \$ =		OR	X \$ =			
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X \$ =		OR	X \$ =			
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))								OR			
	<input type="checkbox"/> 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.						Legal Instrument Examiner: /DEBORAH NASH/						
** 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.												

This collection of information is required by 37 CFR 1.16. 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 12 minutes 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.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



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

NOTICE OF ALLOWANCE AND FEE(S) DUE

95866 7590 12/19/2012
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

EXAMINER

AVERY, JEREMIAH L

ART UNIT PAPER NUMBER

2431

DATE MAILED: 12/19/2012

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/381,219 03/20/2003 David P Yach 13210-1465/KL 9761

TITLE OF INVENTION: SOFTWARE CODE SIGNING SYSTEM AND METHOD

Table with 7 columns: APPLN. TYPE, SMALL ENTITY, ISSUE FEE DUE, PUBLICATION FEE DUE, PREV. PAID ISSUE FEE, TOTAL FEE(S) DUE, DATE DUE
nonprovisional NO \$1770 \$300 \$0 \$2070 03/19/2013

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.
B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

- A. Pay TOTAL FEE(S) DUE shown above, or
B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

**Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

95866 7590 12/19/2012
Fleit Gibbons Gutman Bongini & Bianco P.L.
 551 NW 77th street
 Suite 111
 Boca Raton, FL 33487

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/381,219	03/20/2003	David P Yach	13210-1465/KL	9761

TITLE OF INVENTION: SOFTWARE CODE SIGNING SYSTEM AND METHOD

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1770	\$300	\$0	\$2070	03/19/2013

EXAMINER	ART UNIT	CLASS-SUBCLASS
AVERY, JEREMIAH L	2431	713-001000

<p>1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).</p> <p><input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.</p> <p><input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.</p>	<p>2. For printing on the patent front page, list</p> <p>(1) the names of up to 3 registered patent attorneys or agents OR, alternatively, _____ 1</p> <p>(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. _____ 2</p> <p>_____ 3</p>
---	---

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE _____ (B) RESIDENCE: (CITY and STATE OR COUNTRY) _____

Please check the appropriate assignee category or categories (will not be printed on the patent) : Individual Corporation or other private group entity Government

<p>4a. The following fee(s) are submitted:</p> <p><input type="checkbox"/> Issue Fee</p> <p><input type="checkbox"/> Publication Fee (No small entity discount permitted)</p> <p><input type="checkbox"/> Advance Order - # of Copies _____</p>	<p>4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)</p> <p><input type="checkbox"/> A check is enclosed.</p> <p><input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.</p> <p><input type="checkbox"/> The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).</p>
---	---

5. **Change in Entity Status** (from status indicated above)

a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. 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 12 minutes 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/381,219 03/20/2003 David P Yach 13210-1465/KL 9761

95866 7590 12/19/2012
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487

EXAMINER

AVERY, JEREMIAH L

ART UNIT PAPER NUMBER

2431

DATE MAILED: 12/19/2012

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 1626 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 1626 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)	
	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--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. This communication is responsive to the RCE filed 11/11/11.
2. An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.
3. The allowed claim(s) is/are 166-277. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.
4. 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 the:
 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)).

* Certified copies not received: ____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) 2. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date <u>20111111, 20120202</u> 3. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material 4. <input type="checkbox"/> Interview Summary (PTO-413),
Paper No./Mail Date ____ . | <ol style="list-style-type: none"> 5. <input type="checkbox"/> Examiner's Amendment/Comment 6. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance 7. <input type="checkbox"/> Other ____. |
|--|---|

/Jeremiah Avery/
Examiner, Art Unit 2431

/NATHAN FLYNN/
Supervisory Patent Examiner, Art Unit 2431

Priority

1. This Application, 10/381219, is a national stage entry of PCT/CA01/01344, International Filing Date: 09/20/2001.
2. PCT/CA01/01344 claims priority from Provisional Application 60234152, filed 09/21/2000.

Continued Examination Under 37 CFR 1.114

3. 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 11/11/11 has been entered.

Examiner's Statement of Reasons for Allowance

4. Claims 166-277 are allowed over the prior art.
5. This action is in reply to the applicant's correspondence on 11/11/11.
6. The previous objection to the Applicant's Specification is hereby withdrawn.
7. The following is an examiner's statement of reasons for the indication of allowable claimed subject matter.
8. As per independent claims 166, 191, 216 and 241, generally, the prior art of record, United States Patent No. 6,795,919 to Gibbs et al., and United States Patent No. 6,587,837 to Spagna et al., fails to teach alone, or in combination,

other than via hindsight, at the time of the invention, the features as discussed and remarked upon in the response of 11/11/11.

9. The Applicant's amendments to the claims and presented arguments distinguish the claimed invention over the prior art and place this application in condition for allowance.

Conclusion

10. 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.

11. 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.

12. 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

Application/Control Number: 10/381,219

Page 4

Art Unit: 2431

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 2431

Notice of References Cited	Application/Control No. 10/381,219	Applicant(s)/Patent Under Reexamination YACH ET AL.	
	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-6,895,507	05-2005	Teppler, Steven W.	726/19
*	B	US-6,748,541	06-2004	Margalit et al.	726/9
*	C	US-6,212,636	04-2001	Boyle et al.	713/168
*	D	US-6,697,948	02-2004	Rabin et al.	726/30
*	E	US-6,256,737	07-2001	Bianco et al.	713/186
*	F	US-6,345,256	02-2002	Milsted et al.	705/64
*	G	US-6,374,357	04-2002	Mohammed et al.	726/5
*	H	US-6,587,837	07-2003	Spagna et al.	705/52
*	I	US-6,795,919	09-2004	Gibbs et al.	713/170
	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	Adams, Carlisle. IDUP and SPKM: Developing Public-Key-Based APIs and Mechanisms for Communication Security Services. Proceedings of the Symposium on Network and Distributed System Security. Pub. Date: 1996. Relevant Pages: 128-135. Found on the World Wide Web at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=492419
	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.

Receipt date: 11/11/2011

10381219 - GALL:2431

Doc code: IDS

Approved for use through 07/31/2012. OMB 0651-0031

Doc description: Information Disclosure Statement (IDS) Filed

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		13210-1465	

U.S.PATENTS						Remove
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					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS						Remove
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. Add

FOREIGN PATENT DOCUMENTS								Remove
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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	1541350	CN		2004-10-27			<input type="checkbox"/>
	2	101714201	CN		2011-05-26			<input type="checkbox"/>
	3	101694688	CN		2010-05-26			<input type="checkbox"/>

Receipt date: 11/11/2011 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	13210-1465		

4	1320795	EP		2005-11-16	YACH et al.		<input type="checkbox"/>
5	2306259	EP		2011-04-06	YACH et al.		<input type="checkbox"/>
6	1626324	EP		2006-02-15	YACH et al.		<input type="checkbox"/>
7	2284644	EP		2011-02-16	YACH et al.		<input type="checkbox"/>
8	2278429	EP		2011-01-26	YACH et al.		<input type="checkbox"/>
9	2306260	EP		2011-04-06	YACH et al.		<input type="checkbox"/>
10	1626325	EP		2010-09-01	YACH et al.		<input type="checkbox"/>
11	1626326	EP		2010-09-01	YACH et al.		<input type="checkbox"/>
12	1091666	HK		2007-01-26	YACH et al.	Abstract	<input checked="" type="checkbox"/>
13	1055629	HK		2006-05-04	YACH et al.		<input type="checkbox"/>
14	1091665	HK		2010-11-19	YACH et al.		<input type="checkbox"/>

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	13210-1465		

	15	1091667	HK		2010-11-19	YACH et al.	<input type="checkbox"/>
	16	100573402	CN		2009-12-23		<input type="checkbox"/>

If you wish to add additional Foreign Patent Document citation information please click the Add button **Add**

NON-PATENT LITERATURE DOCUMENTS Remove

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	Notice of Abandonment. Canadian Application No. 2,422,917. Dated: June 20, 2011.	<input type="checkbox"/>
	2	First Office Action. Chinese Application No. 200910207911.0. Dated: August 10, 2011.	<input type="checkbox"/>
	3	Extended European Search Report. European Application No. 10186194.6. Dated: June 22, 2011.	<input type="checkbox"/>
	4	Communication Pursuant to Rules 70(2) and 70a(2) and Reference to Rule 39(1) EPC. European Application No. 10186194.6. Dated: July 25, 2011.	<input type="checkbox"/>
	5	Communication Pursuant to Article 94(3) EPC. European Application No. 10183655.9. Dated: February 23, 2011.	<input type="checkbox"/>
	6	Communication Pursuant to Article 94(3) EPC. European Application No. 10183655.9. Dated: July 13, 2011.	<input type="checkbox"/>
	7	Extended European Search Report (EESR). European Application No. 10183997.5. Dated: December 12, 2010.	<input type="checkbox"/>

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	13210-1465		

8	Communication Pursuant to Article 94(3) EPC. European Application No. 10183997.5. Dated: February 23, 2011.	<input type="checkbox"/>
9	Communication Pursuant to Article 94(3) EPC. European Application No. 10183997.5. Dated: July 14, 2011.	<input type="checkbox"/>
10	Extended European Search Report. European Application No. 10186296.9. Dated: June 22, 2011.	<input type="checkbox"/>
11	Communication Pursuant to Rules 70(2) and 70a(2) and Reference to Rule 39(1) EPC. European Application No. 10186296.9. Dated: July 25, 2011.	<input type="checkbox"/>
12	Invitation pursuant to Article 94(3) and Rule 71(1) EPC dated September 28, 2011, European Patent Application No. 10186296.9.	<input type="checkbox"/>
13	First Office Action. Chinese Application No. 200910209311.8. Dated: October 19, 2011.	<input type="checkbox"/>
14	Chinese Office Action dated September 8, 2011, Chinese Patent Application No. 200910207912.5.	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

EXAMINER SIGNATURE

Examiner Signature	/Jeremiah Avery/	Date Considered	12/14/2012
--------------------	------------------	-----------------	------------

*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	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	13210-1465	

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Kendrick Lo/	Date (YYYY-MM-DD)	2011-11-11
Name/Print	Kendrick Lo	Registration Number	54,948

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./


EAST Search History

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L12	2	((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) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))))).clm.	US-PGPUB; USPAT; UPAD	OR	ON	2012/12/14 08:22

12/ 14/ 2012 8:24:31 AM

C:\ Users\ javery\ Documents\ EAST\ Workspaces\ 10381219.wsp

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	12/14/2012	JLA

SEARCH NOTES		
Search Notes	Date	Examiner
Updated EAST Search	12/14/2012	JLA
Updated Keyword Search within Class 711, subclass 100, Class 713, subclasses 1, 176, 187 and 189Class 395, subclass 682 and Class 719, subclass 328	12/14/2012	JLA
Updated Inventor Search	12/14/2012	JLA
Assignee Search	12/14/2012	JLA
IEEE Search	12/14/2012	JLA
Consulted WQAS with regards to claim 166. Said claim was found to be statutory.	12/14/2012	JLA

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
none	((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) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))))).clm.	12/14/2012	JLA

--	--

Receipt date: 02/02/2012

10381219 - GALL:2431

Doc code: IDS

Doc description: Information Disclosure Statement (IDS) Filed

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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		10381219	
	Filing Date		2003-03-20	
	First Named Inventor	YACH, David P.		
	Art Unit	2431		
	Examiner Name	AVERY, Jeremiah L.		
	Attorney Docket Number	13210-1465/KL		

U.S.PATENTS							Remove	
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							
If you wish to add additional U.S. Patent citation information please click the Add button.							Add	
U.S.PATENT APPLICATION PUBLICATIONS							Remove	
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.							Add	
FOREIGN PATENT DOCUMENTS							Remove	
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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	02/25409	WO		2002-03-28	RESEARCH IN MOTION LIMITED		<input type="checkbox"/>
	2	101694687	CN		2010-04-14	RESEARCH IN MOTION LIMITED		<input type="checkbox"/>
If you wish to add additional Foreign Patent Document citation information please click the Add button							Add	
NON-PATENT LITERATURE DOCUMENTS							Remove	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Receipt date: 02/02/2012		Application Number	10381219	10381219 - GAU: 2431	
			Filing Date	2003-03-20		
			First Named Inventor	YACH, David P.		
			Art Unit	2431		
			Examiner Name	AVERY, Jeremiah L.		
			Attorney Docket Number	13210-1465/KL		

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	Notice of Abandonment. Canadian Application No. 2,422,917. Dated: November 15, 2011.	<input type="checkbox"/>
	2	Notice of Allowance. Canadian Application No. 2,422,917. Dated: September 27, 2010.	<input type="checkbox"/>
	3	Office Action. Canadian Application No. 2,422,917. Dated: March 4, 2009.	<input type="checkbox"/>
	4	Office Action. Canadian Application No. 2,422,917. Dated: March 13, 2008.	<input type="checkbox"/>
	5	Written Opinion. Application No. PCT/CA01/01344. Dated: May 28, 2002.	<input type="checkbox"/>
	6	International Search Report. Application No. PCT/CA01/01344. Dated: April 22, 2002.	<input type="checkbox"/>
	7	Preliminary Examination Report. Application No. PCT/CA01/01344. Dated: November 15, 2002.	<input type="checkbox"/>
	8	Communication under Rule 51(4) EPC. European Application No. 01973901.0. Dated: May 6, 2005.	<input type="checkbox"/>
	9	Communication of a notice of opposition. European Application No. 01973901.0. Dated: August 21, 2006.	<input type="checkbox"/>
	10	Observations to opposition. European Application No. 01973901.0. Dated: May 7, 2007.	<input type="checkbox"/>

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Receipt date: 02/02/2012	Application Number	10381219	10381219 - GAU: 2431
	Filing Date	2003-03-20		
	First Named Inventor	YACH, David P.		
	Art Unit	2431		
	Examiner Name	AVERY, Jeremiah L.		
	Attorney Docket Number	13210-1465/KL		

11	HANDBUCH DER CHIPKARTEN, "Sicherung der Datenubertragung"	<input type="checkbox"/>
12	Summons to attend oral proceedings pursuant to Rule 115(1) EPC. European Application No. 01973901.0. Dated: March 20, 2008.	<input type="checkbox"/>
13	Provision of a copy of the minutes in accordance with Rule 124(4) EPC. European Application No. 01973901.0. Dated: December 22, 2008.	<input type="checkbox"/>
14	Interlocutory decision in Opposition proceedings (Art. 101(3)(a) and 106(2) EPC). European Application No. 01973901.0. Dated: December 22, 2008.	<input type="checkbox"/>
15	First Office Action (English translation). Chinese Application No. 01819200.9. Dated: August 26, 2005.	<input type="checkbox"/>
16	Second Office Action (English translation). Chinese Application No. 01819200.9. Dated: May 30, 2008.	<input type="checkbox"/>
17	Rejection Decision (English translation). Chinese Application No. 01819200.9. Dated: September 26, 2008.	<input type="checkbox"/>
18	Request for Reexamination. Chinese Application No. 01819200.9. Dated: December 24, 2008.	<input type="checkbox"/>
19	Third Office Action (English translation). Chinese Application No. 01819200.9. Dated: April 17, 2009.	<input type="checkbox"/>
20	Certificate of Invention Patent (English translation). Chinese Application No. 01819200.9. Dated: December 23, 2009.	<input type="checkbox"/>
21	Noting of loss of rights pursuant to Rule 112(1) EPC. European Application No. 05024661.0. Dated: December 16, 2011.	<input type="checkbox"/>

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	YACH, David P.	
	Art Unit	2431	
	Examiner Name	AVERY, Jeremiah L.	
	Attorney Docket Number	13210-1465/KL	

22	Communication under Rule 71(3) EPC. European Application No. 05024661.0. Dated: June 29, 2011.	<input type="checkbox"/>
23	Extended European Search Report (EESR). European Application No. 05024661.0. Dated: May 15, 2009.	<input type="checkbox"/>
24	Communication under Rule 71(3) EPC. European Application No. 05024662.8. Dated: February 10, 2010.	<input type="checkbox"/>
25	Extended European Search Report (EESR). European Application No. 05024663.6. Dated: May 15, 2009.	<input type="checkbox"/>
26	Communication under Rule 71(3) EPC. European Application No. 05024663.6. Dated: February 10, 2010.	<input type="checkbox"/>
27	Extended European Search Report (EESR). European Application No. 10183655.9. Dated: December 30, 2010.	<input type="checkbox"/>
28	Extended European Search Report (EESR). European Application No. 10183997.5. Dated: December 21, 2010.	<input type="checkbox"/>
29	ISO/IEC 7816-4 Part 4: "Interindustry commands for interchange" XP002269400	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

EXAMINER SIGNATURE

Examiner Signature	/Jeremiah Avery/	Date Considered	12/14/2012
--------------------	------------------	-----------------	------------

*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	10381219 - GAU: 2431
	Filing Date	2003-03-20	
	First Named Inventor	YACH, David P.	
	Art Unit	2431	
	Examiner Name	AVERY, Jeremiah L.	
	Attorney Docket Number	13210-1465/KL	

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Kendrick Lo/	Date (YYYY-MM-DD)	2012-02-02
Name/Print	Kendrick Lo	Registration Number	54948

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

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	1230	((@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	2012/12/14 08:05
L2	121	(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	2012/12/14 08:05
L3	89	L2 and ((authentic\$ or verify\$ or verification) with signature)	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:05
L4	107983	((@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	2012/12/14 08:12
L5	1130	L4 and ((unauthentic\$ or unverify\$ or unverifi\$) or ("not" near (authentic\$ or verify\$ or verification)))	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:12
L6	578	L5 and (digital near signature)	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:12
L7	575	L6 and access\$	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:12
L8	162	L7 and (API or (application near program\$4 near interface))	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:12
L9	3	L8 and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping)))	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:16
L10	2	(YACH-DAVID-P.in. or BROWN-MICHAEL-	US-	OR	ON	2012/12/14

		S.in. or LITTLE-HERBERT-A.in.) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))).d.m.	US-PGPUB; USPAT; EPO			08:19
L11	3	(research-in-motion-limited.as.) and ((API or (Application near programming near interface)) same (access\$ with (restrict\$ or prohibit\$ or prevent\$ or block\$4 or halt\$3 or deny\$3 or denial or stop or stopping))).d.m.	US-PGPUB; USPAT; EPO	OR	ON	2012/12/14 08:19
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
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-	OR	ON	2009/02/20

			PGPUB; USPAT			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
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	US- PGPUB;	OR	ON	2009/03/05 15:09

		expung\$ or eliminat\$ or eradicat\$ or clear\$5 or remov\$5 or revok\$ or revocat\$) same (software or program or application))	USPAT			
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
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?	US-	OR	ON	2009/03/06

		way or (one near way))))	US-PGPUB; USPAT			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 or 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 (\$crypt\$ 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
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;	OR	ON	2010/08/24 15:06

			EPO			
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;	OR	ON	2010/08/24 15:30

			EPO			
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
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 (@ad<"20000921" @pd<"20000921"	US-PGPUB; USPAT; EPO	OR	ON	2010/08/26 14:16

		@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
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-	OR	ON	2010/08/26

			PGPUB; USPAT; EPO			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
S99	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
S100	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 S99	US- PGPUB; USPAT; EPO	OR	ON	2011/05/06 11:27
S101	78	S100 and ((authentic\$ or verify\$ or verification) with signature)	US- PGPUB; USPAT; EPO	OR	ON	2011/05/06 11:28

12/ 14/ 2012 8:24:40 AM

C:\Users\javery\Documents\EAST\Workspaces\10381219.wsp

SEARCH RESULTS

You searched for: (((API or (Application near programming near interface))) AND access) AND (public or private) and key)

You Refined by:

Publisher: IEEE (x)


Content Type: Conference Publications (x)

Topic: Computing & Processing (Hardware/Software) (x), Components,

Circuits, Devices & Systems (x), Communication, Networking & Broadcasting

(x), Signal Processing & Analysis (x)

Publication Year: 1872 - 2000 (x)

Issue Classification 	Application/Control No. 10381219	Applicant(s)/Patent Under Reexamination YACH ET AL.
	Examiner JEREMIAH AVERY	Art Unit 2431

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant															<input type="checkbox"/> CPA															<input type="checkbox"/> T.D.															<input type="checkbox"/> R.1.47														
17	53	89	125	161	32	197	68	233	104	269																																																	
18	54	90	126	162	33	198	69	234	105	270																																																	
19	55	91	127	163	34	199	70	235	106	271																																																	
20	56	92	128	164	35	200	71	236	107	272																																																	
21	57	93	129	165	36	201	72	237	108	273																																																	
22	58	94	130	1	166	37	202	73	238	109	274																																																
23	59	95	131	2	167	38	203	74	239	110	275																																																
24	60	96	132	3	168	39	204	75	240	111	276																																																
25	61	97	133	4	169	40	205	76	241	112	277																																																
26	62	98	134	5	170	41	206	77	242																																																		
27	63	99	135	6	171	42	207	78	243																																																		
28	64	100	136	7	172	43	208	79	244																																																		
29	65	101	137	8	173	44	209	80	245																																																		
30	66	102	138	9	174	45	210	81	246																																																		
31	67	103	139	10	175	46	211	82	247																																																		
32	68	104	140	11	176	47	212	83	248																																																		
33	69	105	141	12	177	48	213	84	249																																																		
34	70	106	142	13	178	49	214	85	250																																																		
35	71	107	143	14	179	50	215	86	251																																																		
36	72	108	144	15	180	51	216	87	252																																																		

/JEREMIAH AVERY/ Examiner.Art Unit 2431 (Assistant Examiner)	12/14/12 (Date)	Total Claims Allowed: 112	
/NATHAN FLYNN/ Supervisory Patent Examiner.Art Unit 2431 (Primary Examiner)	12/14/2012 (Date)	O.G. Print Claim(s) 166	O.G. Print Figure 3A

Receipt date: 11/11/2011

10381219 - GAU: 2431

Bereskin & Parr

INTELLECTUAL PROPERTY LAW

November 11, 2011

Kendrick Lo B.A.Sc. (Eng. Sci.), MBA, LL.B.
416 957 1685 klo@bereskinparr.com

Your Reference: 10/381,219
Our Reference: 13210-1465/KL

**SUPPLEMENTAL INFORMATION
DISCLOSURE STATEMENT**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA
22313-1450

Dear Sir:

Re: U.S. Patent Application No. 10/381,219
For: SOFTWARE CODE SIGNING SYSTEM AND METHOD
Filing Date: March 20, 2003
Applicants: David P. Yach et al.

In accordance with 37 C.F.R. 1.56 and 1.97(b)(4), the Applicant hereby submits a Supplemental Information Disclosure Statement including (1) a listing, on PTO form SB/08a, of patents and other publications of which the Applicant is aware that may be considered material to patentability, and (2) a copy of foreign and the non-patent literature documents.


The filing of this statement shall be not construed as an admission that the information cited in the attached statement is, or is considered to be, material to patentability (37 CFR 1.97(h)), nor as an admission that it constitutes prior art.

Please have the document recorded against the above-mentioned application.

Respectfully submitted,

BERESKIN & PARR LLP/S.E.N.C.R.L., s.r.l.

By _____


Kendrick Lo
Reg. No. 54,948
(416) 364-7311


Bereskin & Parr LLP

Scotia Plaza, 40 King Street West, 40th Floor, Toronto, Ontario, Canada M5H 3Y2
Tel: 416.364.7311 Fax: 416.363.1398 www.bereskinparr.com

/Jeremiah Avery/ 12/14/2012

TORONTO MISSISSAUGA WATERLOO MONTRÉAL


ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.A./

<i>Index of Claims</i> 	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	12/14/2012									
	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


<input type="checkbox"/> Claims renumbered in the same order as presented by applicant		<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47	
CLAIM		DATE					
Final	Original	12/14/2012					
	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. 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	12/14/2012									
	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. 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	12/14/2012									
	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. 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	12/14/2012									
	145	-									
	146	-									
	147	-									
	148	-									
	149	-									
	150	-									
	151	-									
	152	-									
	153	-									
	154	-									
	155	-									
	156	-									
	157	-									
	158	-									
	159	-									
	160	-									
	161	-									
	162	-									
	163	-									
	164	-									
	165	-									
1	166	=									
2	167	=									
3	168	=									
4	169	=									
5	170	=									
6	171	=									
7	172	=									
8	173	=									
9	174	=									
10	175	=									
11	176	=									
12	177	=									
13	178	=									
14	179	=									
15	180	=									

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	12/14/2012									
16	181	=									
17	182	=									
18	183	=									
19	184	=									
20	185	=									
21	186	=									
22	187	=									
23	188	=									
24	189	=									
25	190	=									
26	191	=									
27	192	=									
28	193	=									
29	194	=									
30	195	=									
31	196	=									
32	197	=									
33	198	=									
34	199	=									
35	200	=									
36	201	=									
37	202	=									
38	203	=									
39	204	=									
40	205	=									
41	206	=									
42	207	=									
43	208	=									
44	209	=									
45	210	=									
46	211	=									
47	212	=									
48	213	=									
49	214	=									
50	215	=									
51	216	=									

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	12/14/2012									
52	217	=									
53	218	=									
54	219	=									
55	220	=									
56	221	=									
57	222	=									
58	223	=									
59	224	=									
60	225	=									
61	226	=									
62	227	=									
63	228	=									
64	229	=									
65	230	=									
66	231	=									
67	232	=									
68	233	=									
69	234	=									
70	235	=									
71	236	=									
72	237	=									
73	238	=									
74	239	=									
75	240	=									
76	241	=									
77	242	=									
78	243	=									
79	244	=									
80	245	=									
81	246	=									
82	247	=									
83	248	=									
84	249	=									
85	250	=									
86	251	=									
87	252	=									

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

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant			<input type="checkbox"/> CPA			<input type="checkbox"/> T.D.			<input type="checkbox"/> R.1.47		
CLAIM			DATE								
Final	Original	12/14/2012									
88	253	=									
89	254	=									
90	255	=									
91	256	=									
92	257	=									
93	258	=									
94	259	=									
95	260	=									
96	261	=									
97	262	=									
98	263	=									
99	264	=									
100	265	=									
101	266	=									
102	267	=									
103	268	=									
104	269	=									
105	270	=									
106	271	=									
107	272	=									
108	273	=									
109	274	=									
110	275	=									
111	276	=									
112	277	=									



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/381,219	03/20/2003	David P Yach	13210-1465/KL

CONFIRMATION NO. 9761

POA ACCEPTANCE LETTER

95866
Fleit Gibbons Gutman Bongini & Bianco P.L.
551 NW 77th street
Suite 111
Boca Raton, FL 33487



Date Mailed: 08/23/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/16/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/atesfai/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/381,219	03/20/2003	David P Yach	13210-1465/KL

CONFIRMATION NO. 9761

POWER OF ATTORNEY NOTICE



89951
Bereskin and Parr LLP
S.E.N.C.R.L., s.r.l.
40 King Street West
40th Floor
Toronto, ON M5H 3Y2
CANADA

Date Mailed: 08/23/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/16/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/atesfai/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

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.

/Jon A. Gibbons/

8/16/2012

Signature

Date

Jon A. Gibbons

Attorney of Record

Printed or Typed Name

Title

This collection of information is required by 37 CFR 3.73(b). 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 1.11 and 1.14. This collection is estimated to take 12 minutes 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt

EFS ID:	13513427
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:	89951
Filer:	Jon A. Gibbons/KAREN TARAGOWSKI
Filer Authorized By:	Jon A. Gibbons
Attorney Docket Number:	13210-1465/KL
Receipt Date:	16-AUG-2012
Filing Date:	20-MAR-2003
Time Stamp:	14:36:09
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	Power of Attorney	10289-US-PCT_POA_8-16-12.pdf	474190 <small>b39675bb7ca2223226bde2610f90967b3a8c579f</small>	no	1

Warnings:

Information:

2	Power of Attorney	10289-US- PCT_sb0096_8-16-12.pdf	422765 2baa7f7a394bbd287ecf41dc26d48588da61d862	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			896955		
<p>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.</p> <p><u>New Applications Under 35 U.S.C. 111</u> 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.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> 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.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> 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.</p>					

Under the Paperwork Reduction Act of 1996, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:

Practitioners associated with the Customer Number: 95866

OR

Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used)

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:

The address associated with Customer Number: 95866

OR

<input type="checkbox"/> Firm or Individual Name	Fleit Gibbons Gutman Bongini & Bianco P.L.		
Address			
City	State	Zip	
Country			
Telephone	Email		

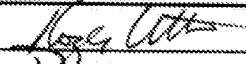
Assignee Name and Address:

Research In Motion Limited
 295 Phillip Street
 Waterloo, Ontario, N2L 3W8, Canada

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	March 9, 2012
Name	Roger Wilheven	Telephone	519-888-7465
Title	Authorized Signing Officer		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. 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 1.11 and 1.14. This collection is estimated to take 3 minutes 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2

Legal OK
 J. H. G. S. P. E. R.
 R. L. + M. R.

RIM OK

03/01/12

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		10381219	
	Filing Date		2003-03-20	
	First Named Inventor	YACH, David P.		
	Art Unit	2431		
	Examiner Name	AVERY, Jeremiah L.		
	Attorney Docket Number	13210-1465/KL		

U.S.PATENTS	Remove
--------------------	--------

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					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS	Remove
--	--------

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. Add

FOREIGN PATENT DOCUMENTS	Remove
---------------------------------	--------

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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	02/25409	WO		2002-03-28	RESEARCH IN MOTION LIMITED		<input type="checkbox"/>
	2	101694687	CN		2010-04-14	RESEARCH IN MOTION LIMITED		<input type="checkbox"/>

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS	Remove
--	--------

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		10381219
	Filing Date		2003-03-20
	First Named Inventor	YACH, David P.	
	Art Unit		2431
	Examiner Name	AVERY, Jeremiah L.	
	Attorney Docket Number		13210-1465/KL

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	Notice of Abandonment. Canadian Application No. 2,422,917. Dated: November 15, 2011.	<input type="checkbox"/>
	2	Notice of Allowance. Canadian Application No. 2,422,917. Dated: September 27, 2010.	<input type="checkbox"/>
	3	Office Action. Canadian Application No. 2,422,917. Dated: March 4, 2009.	<input type="checkbox"/>
	4	Office Action. Canadian Application No. 2,422,917. Dated: March 13, 2008.	<input type="checkbox"/>
	5	Written Opinion. Application No. PCT/CA01/01344. Dated: May 28, 2002.	<input type="checkbox"/>
	6	International Search Report. Application No. PCT/CA01/01344. Dated: April 22, 2002.	<input type="checkbox"/>
	7	Preliminary Examination Report. Application No. PCT/CA01/01344. Dated: November 15, 2002.	<input type="checkbox"/>
	8	Communication under Rule 51(4) EPC. European Application No. 01973901.0. Dated: May 6, 2005.	<input type="checkbox"/>
	9	Communication of a notice of opposition. European Application No. 01973901.0. Dated: August 21, 2006.	<input type="checkbox"/>
	10	Observations to opposition. European Application No. 01973901.0. Dated: May 7, 2007.	<input type="checkbox"/>

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10381219
	Filing Date	2003-03-20
	First Named Inventor	YACH, David P.
	Art Unit	2431
	Examiner Name	AVERY, Jeremiah L.
	Attorney Docket Number	13210-1465/KL

11	HANDBUCH DER CHIPKARTEN, "Sicherung der Datenübertragung"	<input type="checkbox"/>
12	Summons to attend oral proceedings pursuant to Rule 115(1) EPC. European Application No. 01973901.0. Dated: March 20, 2008.	<input type="checkbox"/>
13	Provision of a copy of the minutes in accordance with Rule 124(4) EPC. European Application No. 01973901.0. Dated: December 22, 2008.	<input type="checkbox"/>
14	Interlocutory decision in Opposition proceedings (Art. 101(3)(a) and 106(2) EPC). European Application No. 01973901.0. Dated: December 22, 2008.	<input type="checkbox"/>
15	First Office Action (English translation). Chinese Application No. 01819200.9. Dated: August 26, 2005.	<input type="checkbox"/>
16	Second Office Action (English translation). Chinese Application No. 01819200.9. Dated: May 30, 2008.	<input type="checkbox"/>
17	Rejection Decision (English translation). Chinese Application No. 01819200.9. Dated: September 26, 2008.	<input type="checkbox"/>
18	Request for Reexamination. Chinese Application No. 01819200.9. Dated: December 24, 2008.	<input type="checkbox"/>
19	Third Office Action (English translation). Chinese Application No. 01819200.9. Dated: April 17, 2009.	<input type="checkbox"/>
20	Certificate of Invention Patent (English translation). Chinese Application No. 01819200.9. Dated: December 23, 2009.	<input type="checkbox"/>
21	Noting of loss of rights pursuant to Rule 112(1) EPC. European Application No. 05024661.0. Dated: December 16, 2011.	<input type="checkbox"/>

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		10381219
	Filing Date		2003-03-20
	First Named Inventor	YACH, David P.	
	Art Unit	2431	
	Examiner Name	AVERY, Jeremiah L.	
	Attorney Docket Number	13210-1465/KL	

22	Communication under Rule 71(3) EPC. European Application No. 05024661.0. Dated: June 29, 2011.	<input type="checkbox"/>
23	Extended European Search Report (EESR). European Application No. 05024661.0. Dated: May 15, 2009.	<input type="checkbox"/>
24	Communication under Rule 71(3) EPC. European Application No. 05024662.8. Dated: February 10, 2010.	<input type="checkbox"/>
25	Extended European Search Report (EESR). European Application No. 05024663.6. Dated: May 15, 2009.	<input type="checkbox"/>
26	Communication under Rule 71(3) EPC. European Application No. 05024663.6. Dated: February 10, 2010.	<input type="checkbox"/>
27	Extended European Search Report (EESR). European Application No. 10183655.9. Dated: December 30, 2010.	<input type="checkbox"/>
28	Extended European Search Report (EESR). European Application No. 10183997.5. Dated: December 21, 2010.	<input type="checkbox"/>
29	ISO/IEC 7816-4 Part 4: "Interindustry commands for interchange" XP002269400	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

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	YACH, David P.
	Art Unit	2431
	Examiner Name	AVERY, Jeremiah L.
	Attorney Docket Number	13210-1465/KL

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Kendrick Lo/	Date (YYYY-MM-DD)	2012-02-02
Name/Print	Kendrick Lo	Registration Number	54948

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 March 2002 (28.03.2002)

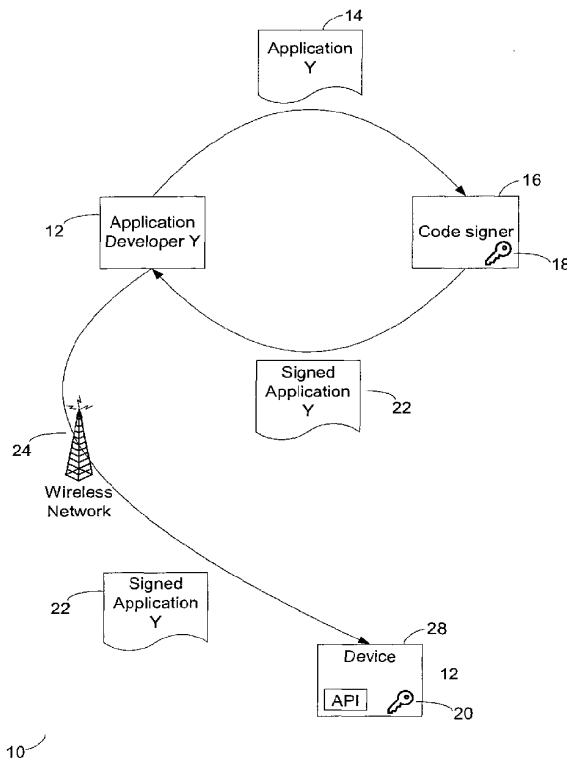
PCT

(10) International Publication Number
WO 02/25409 A2

- (51) International Patent Classification⁷: **G06F 1/00**
- (21) International Application Number: PCT/CA01/01344
- (22) International Filing Date:
20 September 2001 (20.09.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/234,152 21 September 2000 (21.09.2000) US
60/235,354 26 September 2000 (26.09.2000) US
60/270,663 20 February 2001 (20.02.2001) US
- (71) Applicant (for all designated States except US): **RESEARCH IN MOTION LIMITED** [CA/CA]; 295 Phillip Street, Waterloo, Ontario N2L 3W8 (CA).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **YACH, David, P.** [CA/CA]; 254 Castlefield Avenue, Waterloo, Ontario N2K 2N1 (CA). **BROWN, Michael, S.** [CA/CA]; 7 Danube Street, Heidelberg, Ontario N0B 1Y0 (CA). **LITTLE, Herbert, A.** [CA/CA]; 504 Old Oak Place, Waterloo, Ontario N2T 2V8 (CA).
- (74) Agent: **PATHIVAL, Krishna, K.**; Research In Motion Limited, 295 Phillip Street, Waterloo, Ontario N2L 3W8 (CA).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK,

[Continued on next page]

(54) Title: CODE SIGNING SYSTEM AND METHOD



(57) Abstract: 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 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.

WO 02/25409 A2



SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Code Signing System And Method

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from and is related to the following prior applications:

- 5 "Code Signing System And Method," United States Provisional Application No. 60/234,152, filed September 21, 2000; "Code Signing System And Method," United States Provisional Application No. 60/235,354, filed September 26, 2000; and "Code Signing System And Method," United States Provisional Application No. 60/270,663, filed February 20, 2001.

10

BACKGROUND

1. FIELD OF THE INVENTION

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™ applications for mobile communication devices, such as
15 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

Security protocols involving software code signing schemes are known. Typically, such
20 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
5 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.

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.

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

SUMMARY

A code signing system and method is provided. The code signing system operates in
15 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.

20 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.

5 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
10 sensitive API, then denying the software application access to the sensitive API.

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
15 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.

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
20 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.

5 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
10 the software application access to the one or more APIs.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram illustrating a code signing protocol according to one embodiment of the invention;

15 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
20 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

5 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
10 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
15 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
20 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.

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 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 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.

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
5 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.

The binary executable representation of software application Y 14 may be independent of
10 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
15 software application Y 14 targets several mobile devices.

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
20 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.

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.

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
5 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.

The public signature key 20 corresponds to the private signature key 18 maintained by
10 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
15 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
20 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.

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.

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.

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
5 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
10 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.

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
15 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
20 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.

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.

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™ (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.

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.

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.

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.

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

5 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
10 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.

In step 106, the virtual machine retrieves the public signature key 20 and signature
15 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
20 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.

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.

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
5 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.

At step 220, a code signing authority for one target device receives a target-signing
10 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
15 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
20 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.

5 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.

10 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.

15 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
20 verifications, for instance periodically or when a new revocation list is downloaded.

 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.

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).

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™ mobile communication system or DataTAC™ mobile communication system, whereas a device 610 intended for use in Europe may incorporate a General Packet Radio Service (GPRS) communication subsystem 611.

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.

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.

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
5 612 and transmitter 614 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 620.

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
10 638 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
15 access is granted, may be provided on the device 610 to interface between software applications and any of the resources shown in Fig. 6.

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
20 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.

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. 5 It is contemplated that received and transmitted communication signals may also be stored to RAM 626.

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 10 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. 15 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 20 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 628, 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).

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.

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.

5 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
10 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.

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
15 need not necessarily be similar devices. For example, the subsystem 640 may include an infrared device and associated circuits and components or a Bluetooth™ communication module to provide for communication with similarly-enabled systems and devices.

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
20 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.

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.

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.

We claim:

1. A code signing system for operation in conjunction with a software application having a digital signature, comprising:
 - an application platform;
 - 5 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.
- 10 2. The code signing system of claim 1, wherein the virtual machine denies the software application access to the API if the digital signature is not authentic.
3. The code signing system of claim 1, wherein the virtual machine purges the software application if the digital signature is not authentic.
- 15 4. The code signing system of claim 1, wherein the code signing system is installed on a mobile device.
5. The code signing system of claim 1, wherein the digital signature is generated by a code
20 signing authority.

6. A code signing system for operation in conjunction with a software application having a digital signature, comprising:

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.

10

7. The code signing system of claim 6, wherein the plurality of APIs are included in an API library.

8. The code signing system of claim 6, wherein one or more of the plurality of APIs is classified as sensitive, and wherein the virtual machine uses the digital signature to control access to the sensitive APIs.

9. The code signing system of claim 8, for operation in conjunction with a plurality of software applications, wherein one or more of the plurality of software applications has a digital signature, 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 in order to control access to the sensitive APIs by each of the plurality of software applications.

10. The code signing system of claim 6, wherein the resource on the application platform comprises a wireless communication system.
- 5 11. The code signing system of claim 6, wherein the resource on the application platform comprises a cryptographic module which implements cryptographic algorithms.
12. The code signing system of claim 6, wherein the resource on the application platform comprises a data store.
- 10 13. The code signing system of claim 6, wherein the resource on the application platform comprises a user interface (UI).
14. The code signing system of claim 1, further comprising:
- 15 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.
15. The code signing system of claim 14, wherein one or more of the plurality of API libraries is classified as sensitive, and wherein the virtual machine uses the digital signature to control
- 20 access to the sensitive API libraries by the software application.

16. The code signing system of claim 15, wherein the software application includes a unique digital signature for each sensitive API library.

17. The code signing system of claim 16, wherein:

5 the software application includes a signature identification for each unique digital signature;

 each sensitive API library includes a signature identifier; and

 the virtual machine compares the signature identification and the signature identifier to match the unique digital signatures with sensitive API libraries.

10

18. The code signing system of claim 1, 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.

15 19. The code signing system of claim 18, 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.

20

20. The code signing system of claim 1, wherein the API further comprises:
a description string that is displayed by the mobile device when the software application attempts to access the API.
- 5 21. The code signing system of claim 1, wherein the application platform comprises an operating system.
22. The code signing system of claim 1, wherein the application platform comprises one or more core functions of a mobile device.
- 10 23. The code signing system of claim 1, wherein the application platform comprises hardware on a mobile device.
24. The code signing system of claim 23, wherein the hardware comprises a subscriber identity
15 module (SIM) card.
25. The code signing system of claim 1, wherein the software application is a Java application for a mobile device.
- 20 26. The code signing system of claim 1, wherein the API interfaces with a cryptographic routine on the application platform.

27. The code signing system of claim 1, wherein the API interfaces with a proprietary data model on the application platform.
28. The code signing system of claim 1, wherein the virtual machine is a Java virtual machine
5 installed on a mobile device.
29. 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
10 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.
- 15
30. The method of claim 29, comprising 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.
- 20
31. The method of claim 29, wherein the digital signature is generated by a code signing authority.

32. The method of claim 29, comprising 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

5 if the digital signature is not authentic, then denying the software application access to
the sensitive API.

33. The method of claim 32, comprising the additional step of:

if the digital signature is not authentic, then purging the software application from the
mobile device.

10

34. The method of claim 32, 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:

15 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.

20 35. The method of claim 34, wherein the digital signature is generated by calculating a hash of
the software application and applying the private signature key.

36. The method of claim 29, 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.

5 37. The method of claim 36, comprising the additional step of:

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

38. A method of controlling access to an application programming interface (API) on a mobile
10 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;

15 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.

39. The method of claim 38, wherein the step of reviewing the software application is performed
20 by a code signing authority.

40. The method of claim 38, wherein the step of appending the digital signature to the software application is 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

5 signature.

41. The method of claim 40, wherein the hash of the software application is calculated using the Secure Hash Algorithm (SHA1).

10 42. The method of claim 40, wherein the step of verifying the authenticity of a 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;

15 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.

20 43. The method of claim 42, comprising the further step of, if the digital signature is not
authentic, then denying the software application access to the API.

44. The method of claim 42, wherein the signature key is a private signature key and the corresponding signature key is a public signature key.

45. A method of controlling access to a sensitive application programming interface (API) on a
5 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
10 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

15 the mobile device verifies the authenticity of the digital signature in order to control access to the sensitive API by the software application.

46. The method of claim 45, wherein the step of generating the digital signature is performed by a code signing authority.

20

47. The method of claim 45, wherein the step of generating the digital signature is performed by applying a signature key to the hash of the software application.

48. The method of claim 47, 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;
 - 5 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
 - 10 recovered hash; and
 - if the digital signature is not authentic, then denying the software application access to the sensitive API.
49. A method of restricting access to application programming interfaces on a mobile device,
- 15 comprising 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
 - 20 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.

50. The method of claim 49, comprising 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.
- 5 51. The method of claim 49, wherein:
the software application includes a plurality of digital signatures; and
the plurality of digital signatures includes digital signatures respectively associated with different types of mobile devices.
- 10 52. The method of claim 51, wherein each of the plurality of digital signatures is generated by a respective corresponding code signing authority.
53. The method of claim 49, wherein the step of determining whether or not the software application includes an authentic digital signature associated with the mobile device comprises
15 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.
- 20 54. The method of claim 53, wherein the one or more APIs includes one or more APIs classified as sensitive, and the method further comprises 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.

55. The method of claim 52, wherein each of the plurality of digital signatures is generated by
5 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.

56. The method of claim 55, wherein the step of determining whether or not the software application includes an authentic digital signature associated with the mobile device comprises
10 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
15 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;
20 applying the public signature key to the digital signature to obtain a recovered hash; and comparing the generated hash with the recovered hash.

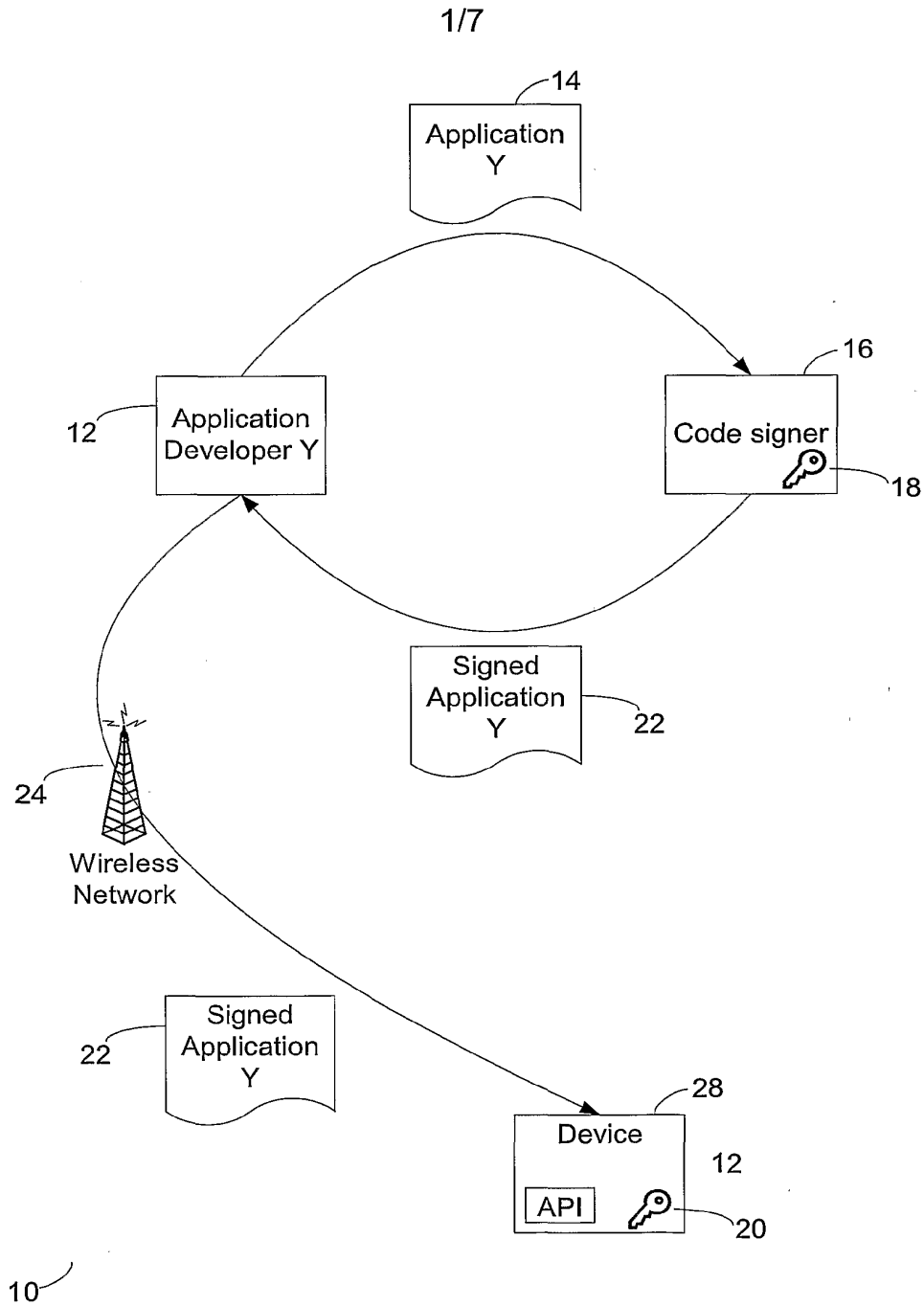
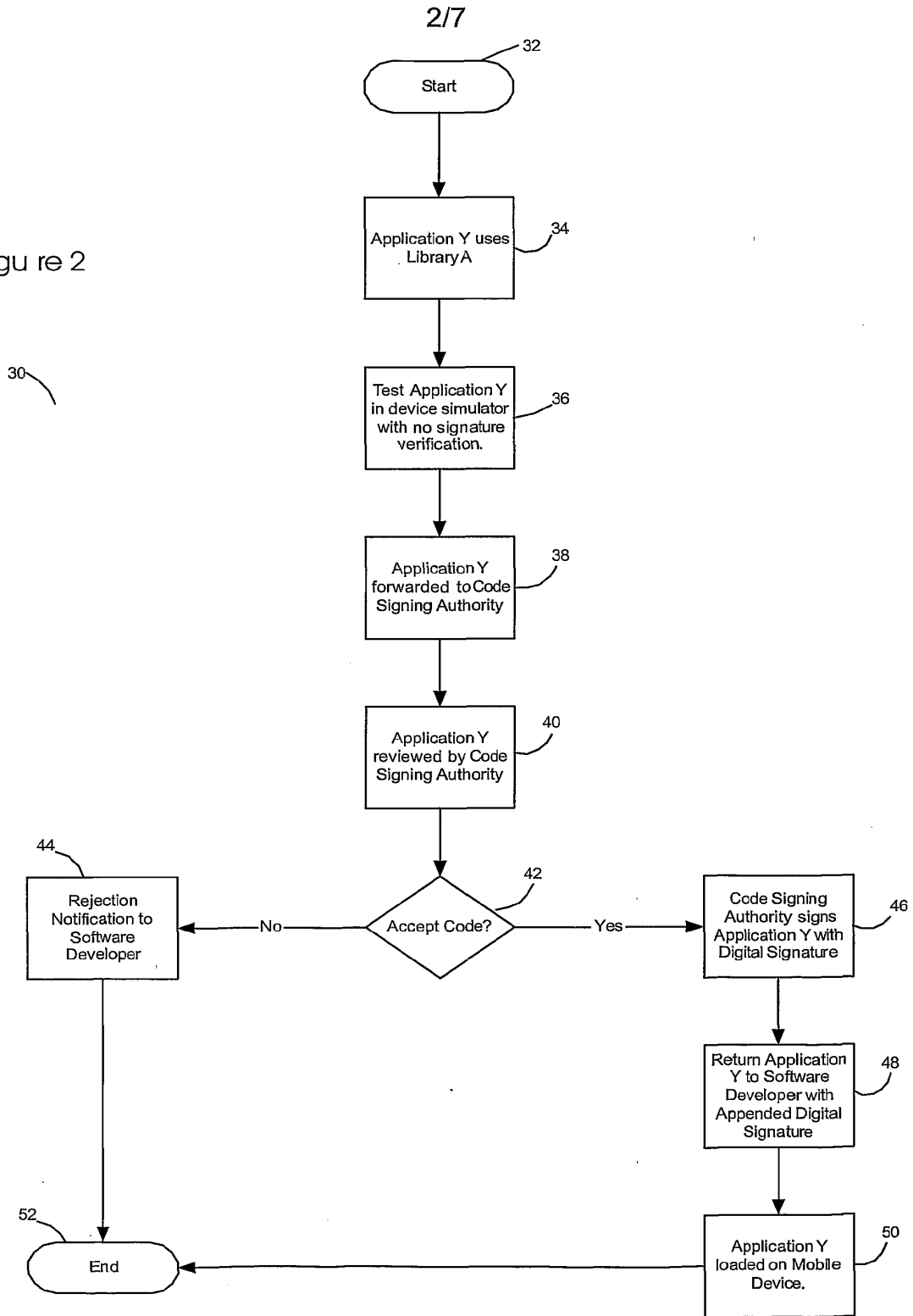


Figure 1

Figure 2



3/7

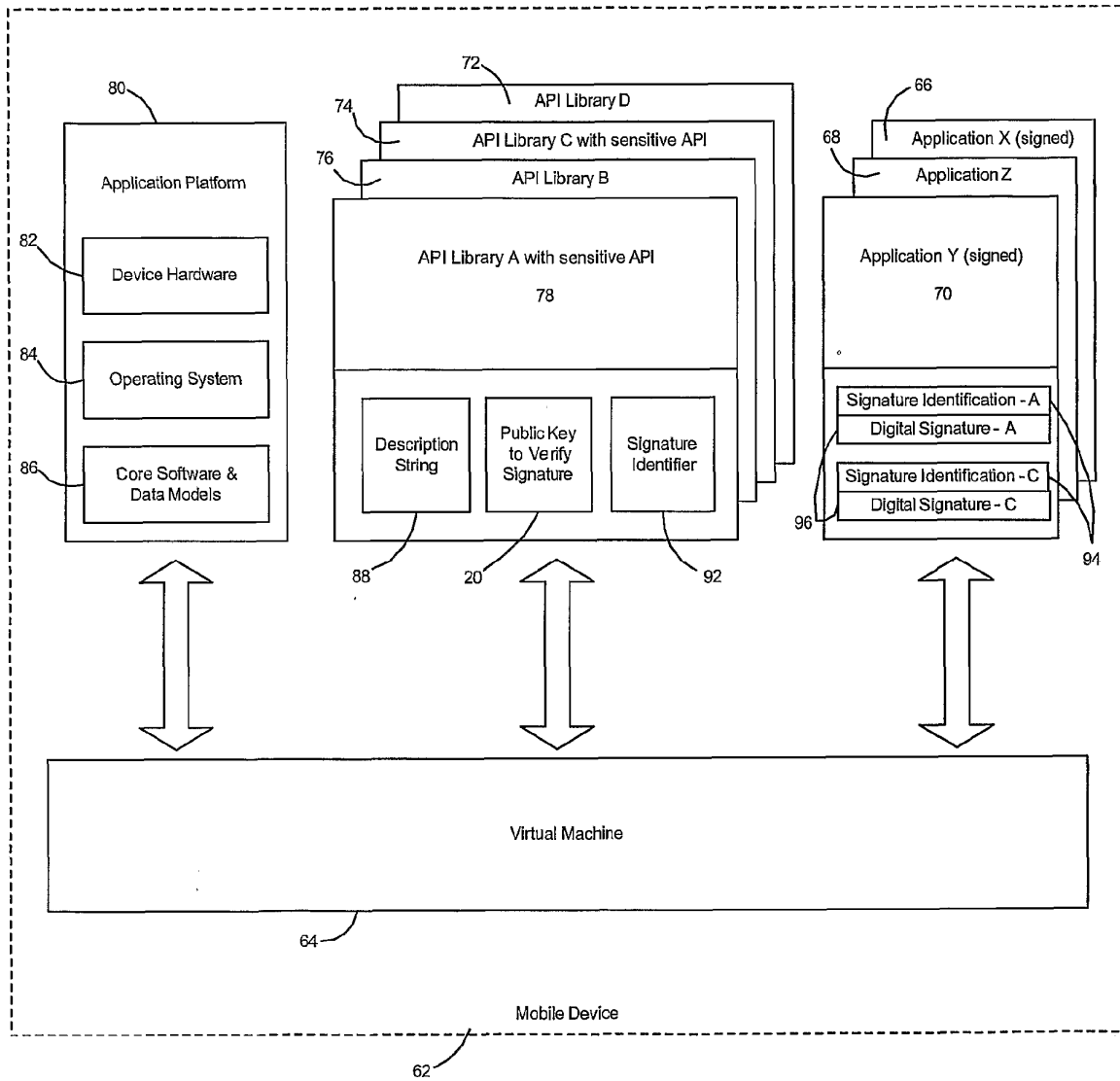


Figure 3

4/7

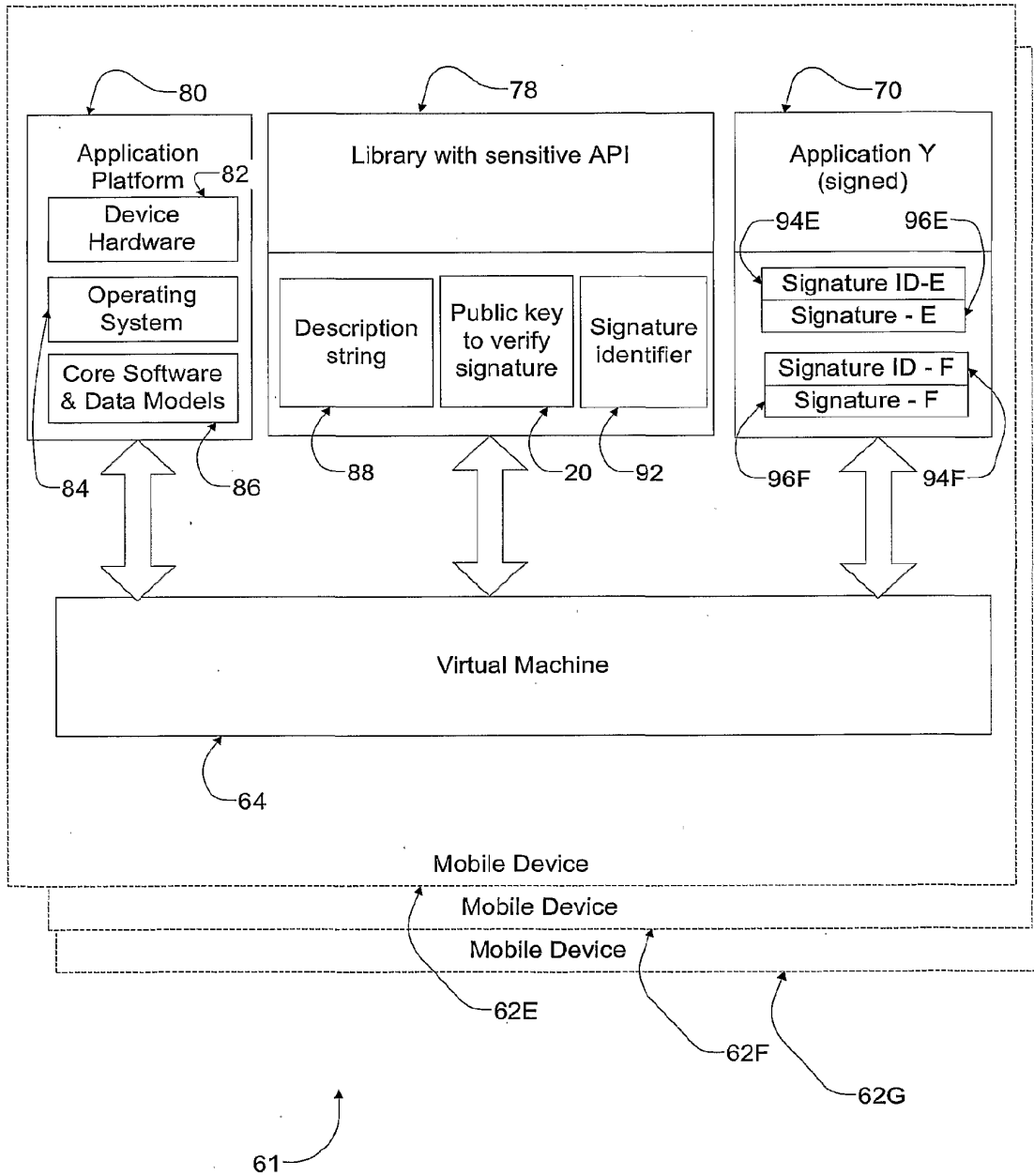
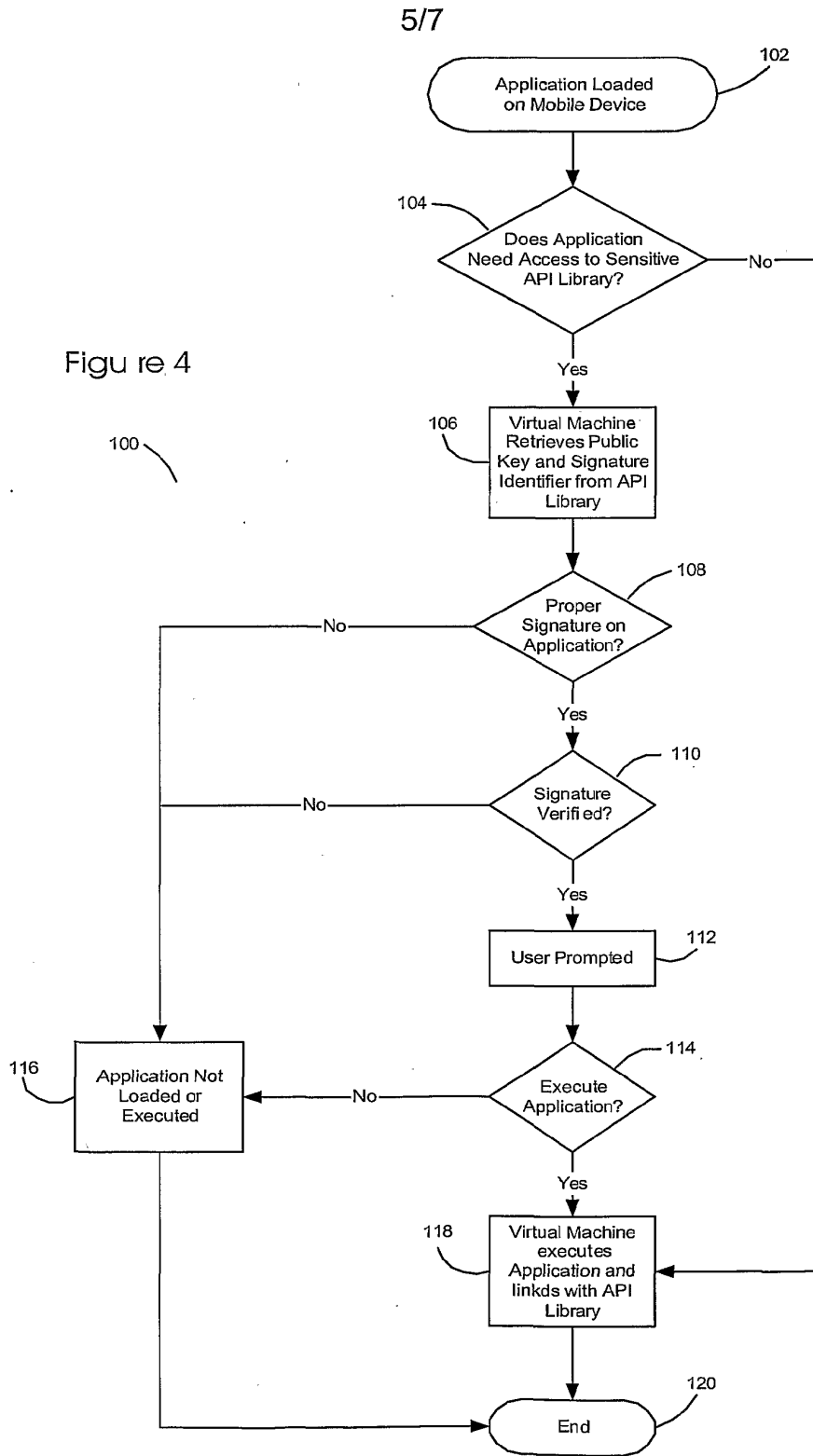


Figure 3A

Figure 4



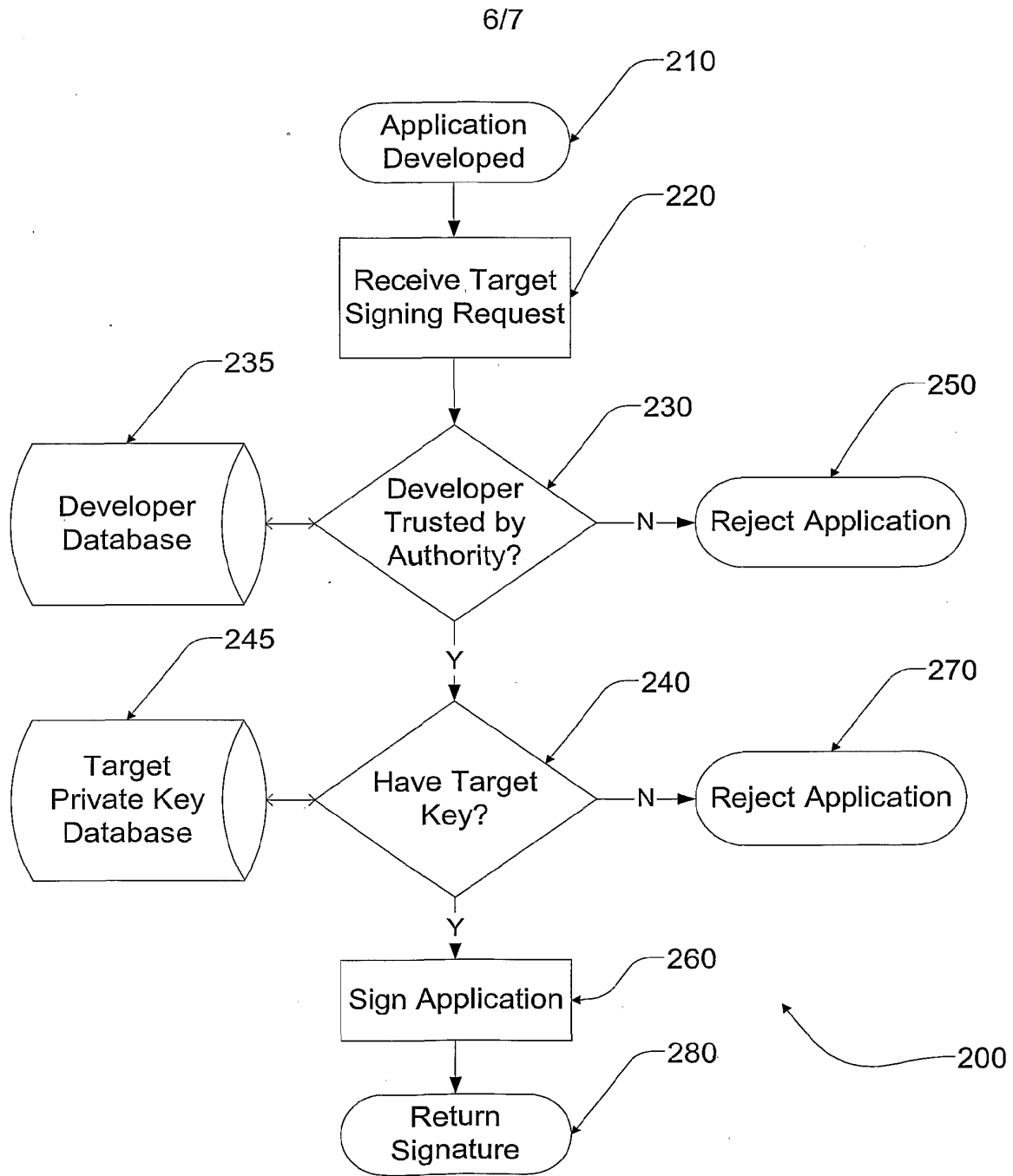


Figure 5

7/7

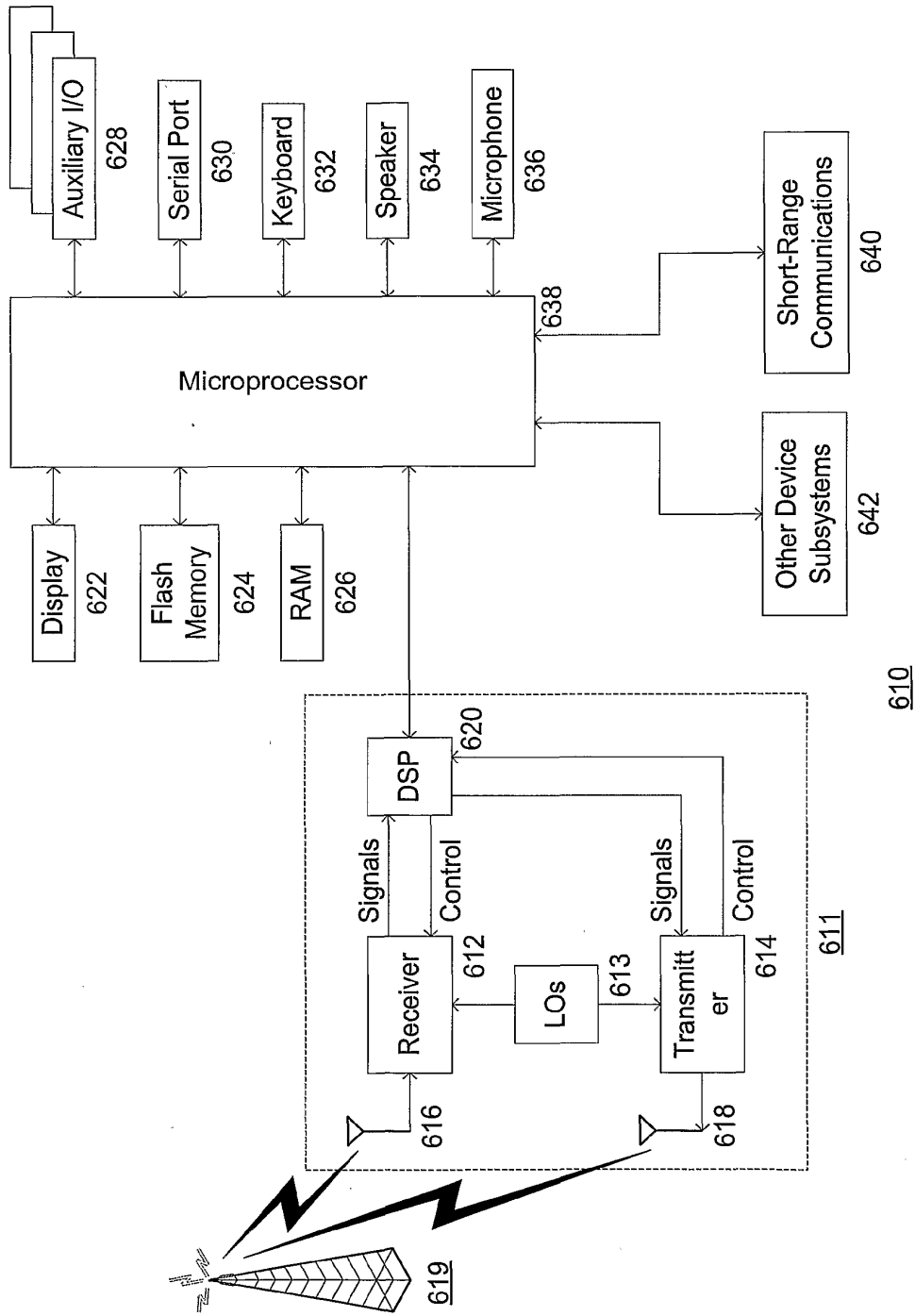


Figure 6



(12) 发明专利申请

(10) 申请公布号 CN 101694687 A

(43) 申请公布日 2010.04.14

(21) 申请号 200910207911.0

(22) 申请日 2001.09.20

(30) 优先权数据

60/234,152 2000.09.21 US

60/235,354 2000.09.26 US

60/270,663 2001.02.20 US

(62) 分案原申请数据

01819200.9 2001.09.20

(71) 申请人 捷讯研究有限公司

地址 加拿大安大略省沃特卢市

(72) 发明人 戴维·P·亚切 迈克尔斯·S·布朗

赫伯特·A·利特尔

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 戎志敏

(51) Int. Cl.

G06F 21/22 (2006.01)

H04L 29/06 (2006.01)

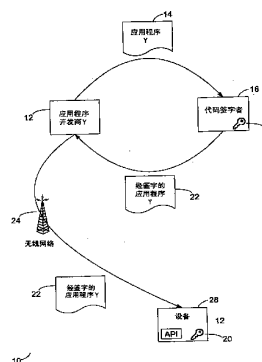
权利要求书 4 页 说明书 14 页 附图 7 页

(54) 发明名称

代码签字系统和方法

(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。



CN 101694687 A

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;

虚拟机验证数字签字的真实性,以便控制软件应用程序访问 API,其中,签字标识符对应签字标识,

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:

(i) 如果数字签字不真实,则虚拟机拒绝软件应用程序访问 API;

或

(ii) 如果数字签字不真实,则虚拟机删除软件应用程序。

3. 根据权利要求 2 所述的代码签字系统,其特征在于:

(iii) 代码签字系统装在移动设备上;

或

(iv) 数字签字由代码签字授权机构产生。

4. 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。

5. 根据权利要求 1 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。

6. 根据权利要求 3 所述的代码签字系统,其特征在于敏感的 API 程序库还包括当软件应用程序试图访问敏感的 API 时,移动显示描述字符串。

7. 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;

或一个或多个移动设备的核心功能;

或

移动设备上的硬件。

8. 根据权利要求 7 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。

9. 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。

10. 根据权利要求 1 所述的代码签字系统,其特征在于

(i)API 与应用平台上的加密流程接口;

或

(ii)API 与应用平台上的专用数据模块接口。

11. 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。

12. 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,包括步骤:

把软件应用程序装载到移动设备上,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。

13. 根据权利要求 12 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。

14. 根据权利要求 12 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。

15. 根据权利要求 12 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。

16. 根据权利要求 15 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。

17. 根据权利要求 12 所述的方法,其特征在于当软件应用程序试图访问所述至少一个敏感的 API 时,向用户显示描述字符串。

18. 根据权利要求 12 所述的方法,其特征在于还包括步骤:

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。

19. 根据权利要求 18 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。

20. 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载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,则授权数字签字;

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。

21. 根据权利要求 20 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。

22. 根据权利要求 20 所述的方法,其特征在于还包括步骤:

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软件应用程序。

23. 根据权利要求 20 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不同类型的移动设备有关的数字签字和签字标识。

24. 根据权利要求 23 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。

25. 根据权利要求 20 所述的方法,其特征在于还包括确定软件应用程序是否包括真实的数字签字,其中,如果签字标识与至少一个签字标识符对应,则验证数字签字的真实性。

26. 根据权利要求 24 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。

27. 根据权利要求 20 所述的方法,其特征在于所述如果签字标识对应签字标识符则授权数字签字包括:

验证与授权数字签字的签字标识符对应的签字标识,其中,签字标识与签字标识符对应包括:

在移动设备上存储公用签字密钥,所述公用密钥与产生数字签字的代码签字授权机构关联的专用签字密钥对应;

产生软件应用程序的杂乱信号,获得产生的杂乱信号;

将公用签字密钥应用到数字签字,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。

28. 根据权利要求 20 所述的方法,其特征在于:

移动设备包括多个 API;

至少一个 API 被分类为敏感的;

访问任一个 API 需要真实的全局签字;

访问每一个敏感的 API 需要真实的全局签字和与签字标识符关联的真实的数字签字;

确定软件应用程序包括真实的数字签字和签字标识符包括:

确定软件应用程序要求访问的一个或多个 API 是否包括敏感的 API;

确定软件应用程序是否包括真实的全局签字;

确定软件应用程序是否包括真实的数字签字和签字标识符,其中,软件应用程序要求访问的一个或多个 API 包括敏感的 API,并且软件应用程序真实的全局签字;

拒绝软件应用程序访问一个或多个 API 包括:

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

拒绝软件应用程序访问敏感的 API,其中,软件应用程序要求访问的一个或多个 API 包

括密感的 API, 软件应用程序包括真实的全局签字, 但软件应用程序不包括要求访问密感的 API 的真实的数字签字和签字标识符。

代码签字系统及方法

[0001] 有关申请的参照

[0002] 本申请要求下列申请的优先权：

[0003] “代码签字系统及方法”于2000年9月21日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日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日申请的美国临时申请，申请号是60/270663；

技术领域

[0004] 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™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

[0005] 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者的软件应用程序。一旦该软件被用户下载，用户必须只根据对软件开发者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。

[0006] 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。

[0007] 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。

发明内容

[0008] 本发明的目的是提供代码签字系统和方法。

[0009] 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：

[0010] 应用平台；

[0011] 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；

[0012] 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，

[0013] 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。

- [0014] 优选地,如果数字签字不真实,则虚拟机拒绝软件应用程序访问 API。
- [0015] 优选地,如果数字签字不真实,则虚拟机删除软件应用程序。
- [0016] 优选地,代码签字系统装在移动设备上。
- [0017] 优选地,数字签字由代码签字授权机构产生。
- [0018] 优选地,还包括:
- [0019] 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。
- [0020] 优选地,至少一个 API 程序库被分类为敏感的;
- [0021] 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;
- [0022] 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,用于访问敏感的 API 程序库;
- [0023] 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。
- [0024] 优选地,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。
- [0025] 优选地,应用平台包括操作系统。
- [0026] 优选地,包括一个或多个移动设备的核心功能。
- [0027] 优选地,包括移动设备上的硬件。
- [0028] 优选地,硬件包括用户身份模块卡。
- [0029] 优选地,软件应用程序是用于移动设备的 Java 应用程序。
- [0030] 优选地,API 与应用平台上的加密流程接口。
- [0031] 优选地,API 与应用平台上的专用数据模块接口。
- [0032] 优选地,虚拟机是安装在移动设备上的 Java 虚拟机。
- [0033] 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:
 - [0034] 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;
 - [0035] 确定软件应用程序是否包括数字签字和签字标识;
 - [0036] 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;
 - [0037] 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,
 - [0038] 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:
 - [0039] 在移动设备上存储对应专用签字密钥的公用签字密钥;
 - [0040] 产生软件应用程序的杂乱信号来获得产生的杂乱信号;
 - [0041] 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;

- [0042] 比较产生的杂乱信号和恢复的杂乱信号。
- [0043] 优选地,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。
- [0044] 优选地,数字签字和签字标识由代码签字授权机构产生。
- [0045] 优选地,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。
- [0046] 优选地,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。
- [0047] 优选地,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。
- [0048] 优选地,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。
- [0049] 优选地,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。
- [0050] 按照本发明的另一方面,一种移动设备,包括:
- [0051] 应用平台,具有应用编程接口 API;
- [0052] 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;
- [0053] 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;
- [0054] 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,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设备;
- [0055] 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;
- [0056] 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;
- [0057] 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;
- [0058] 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,
- [0059] 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。
- [0060] 优选地,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。
- [0061] 优选地,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。
- [0062] 优选地,应用平台的 API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0063] 优选地,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。

- [0064] 优选地,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。
- [0065] 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:
- [0066] 从软件开发商接收软件应用程序;
- [0067] 确定软件应用程序是否满足至少一个标准;
- [0068] 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;
- [0069] 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;
- [0070] 如果数字签字是真实的,向软件应用程序提供到 API 的路径;
- [0071] 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:
- [0072] 计算软件应用程序的杂乱信号;
- [0073] 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;
- [0074] 在移动设备上提供公用签字密钥;
- [0075] 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;
- [0076] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0077] 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。
- [0078] 优选地,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。
- [0079] 优选地,使用安全的杂乱信号算法计算软件应用程序的杂乱信号。
- [0080] 优选地,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。
- [0081] 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程序编程接口 API 的方法,包括步骤:
- [0082] 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;
- [0083] 接收软件应用程序的杂乱信号;
- [0084] 确定杂乱信号是否是注册的软件开发商所发送;
- [0085] 产生数字签字,其中,
- [0086] 数字签字和签字标识被添加到软件应用程序;
- [0087] 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;
- [0088] 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;
- [0089] 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:
- [0090] 在移动设备上提供公用签字密钥;
- [0091] 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;
- [0092] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0093] 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;

- [0094] 如果数字签字不真实,则拒绝软件应用程序访问敏感的 API。
- [0095] 优选地,产生数字签字的步骤由代码签字授权机构执行。
- [0096] 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:
- [0097] 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;
- [0098] 如果签字标识对应签字标识符,则验证数字签字;
- [0099] 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;
- [0100] 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:
- [0101] 验证与签字标识符对应的签字标识;
- [0102] 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;
- [0103] 产生软件应用程序的杂乱信号,以获得产生的杂乱信号;
- [0104] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0105] 将产生的杂乱信号与恢复的杂乱信号进行比较。
- [0106] 优选地,数字签字和签字标识与移动设备的类型有关。
- [0107] 优选地,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。
- [0108] 优选地,软件应用程序包括多个数字签字和签字标识;
- [0109] 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。
- [0110] 优选地,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。
- [0111] 优选地,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。
- [0112] 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,软件应用程序具有数字签字和签字标识,包括:
- [0113] 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;
- [0114] 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;
- [0115] 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;
- [0116] 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;
- [0117] 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。
- [0118] 优选地,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。
- [0119] 优选地,API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0120] 优选地,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。

[0121] 优选地,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。

[0122] 优选地,要求软件应用程序提供全局数字签字的验证,以访问任何 API。

附图说明

[0123] 图 1 是根据本发明实施例的代码签字协议图;

[0124] 图 2 是图 1 的代码签字协议的流程图;

[0125] 图 3 是在移动设备上的代码签字系统方框图;

[0126] 图 3A 是在一组移动设备上的代码签字系统方框图;

[0127] 图 4 是图 3 和图 3A 代码签字系统的工作流程图;

[0128] 图 5 是管理图 3A 的代码签字真实性的流程图;

[0129] 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

[0130] 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 Java 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。

[0131] 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(如地址簿或日历本)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。

[0132] 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职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。

[0133] 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。

[0134] 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)。

[0135] 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,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 SHA1)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。

[0136] 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。

[0137] 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。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。

[0138] 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。

[0139] 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。

[0140] 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附加到软件应用程序 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。

[0141] 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。

[0142] 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,或分配到公司雇员。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。

[0143] 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。

[0144] 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。

[0145] 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,并能使虚拟机让数字签字 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。

[0146] 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 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。

[0147] 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 程序库 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,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。

[0148] 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。

[0149] 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。

[0150] 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)。

[0151] 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。

[0152] 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。

[0153] 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。

[0154] 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。

[0155] 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。

[0156] 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)。

[0157] 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,本地振荡器(L0)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™ 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。

[0158] 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。

[0159] 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。

[0160] 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。

[0161] 设备 610 优先包括微处理器 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。

[0162] 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 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。

[0163] 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。

[0164] 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。

[0165] 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。

[0166] 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。

[0167] 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。

[0168] 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 Bluetooth™(蓝牙)通信模式,以提供与有相同能力的系统和设备通信。

[0169] 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。

[0170] 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。

[0171] 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。

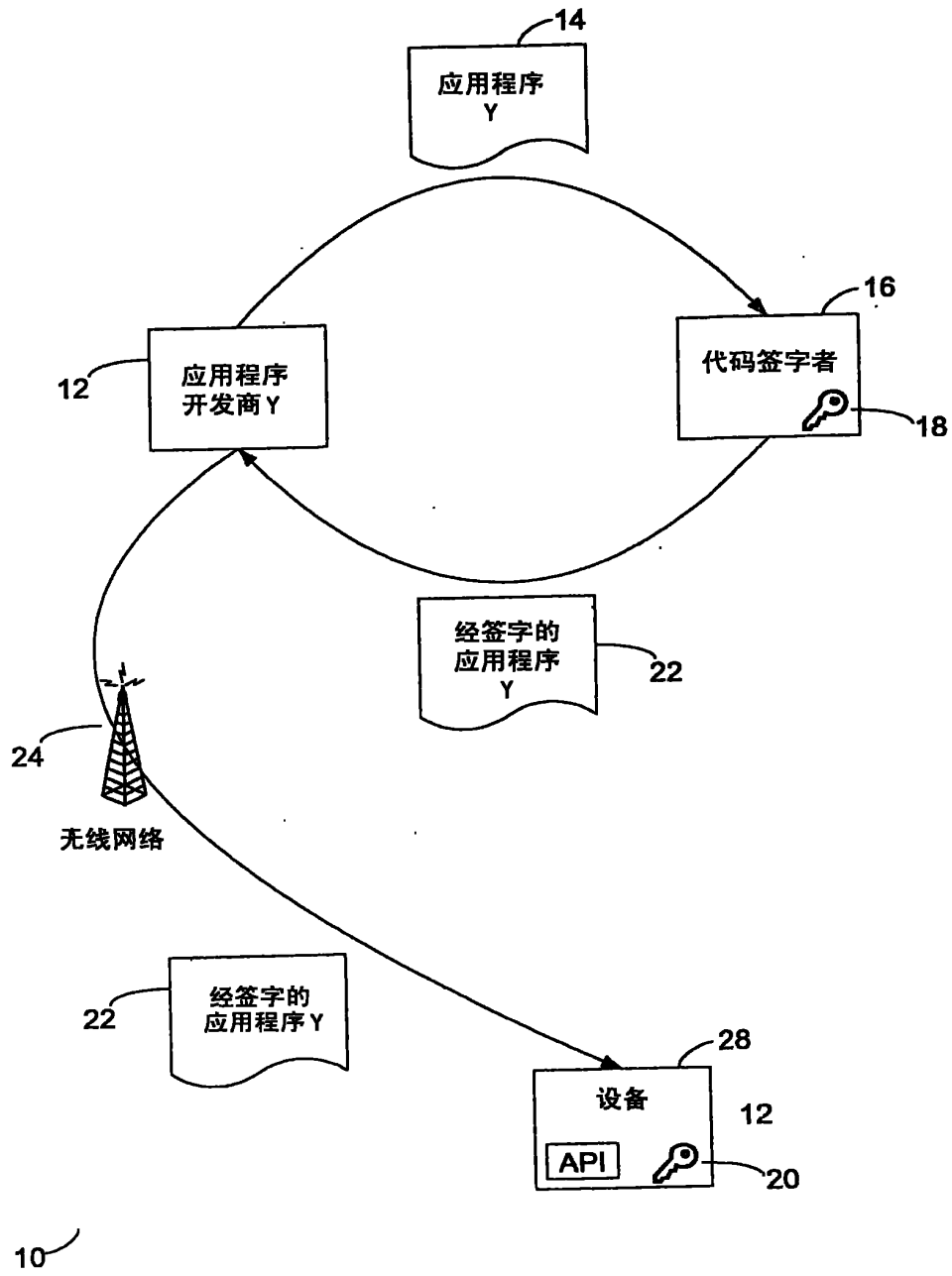


图 1

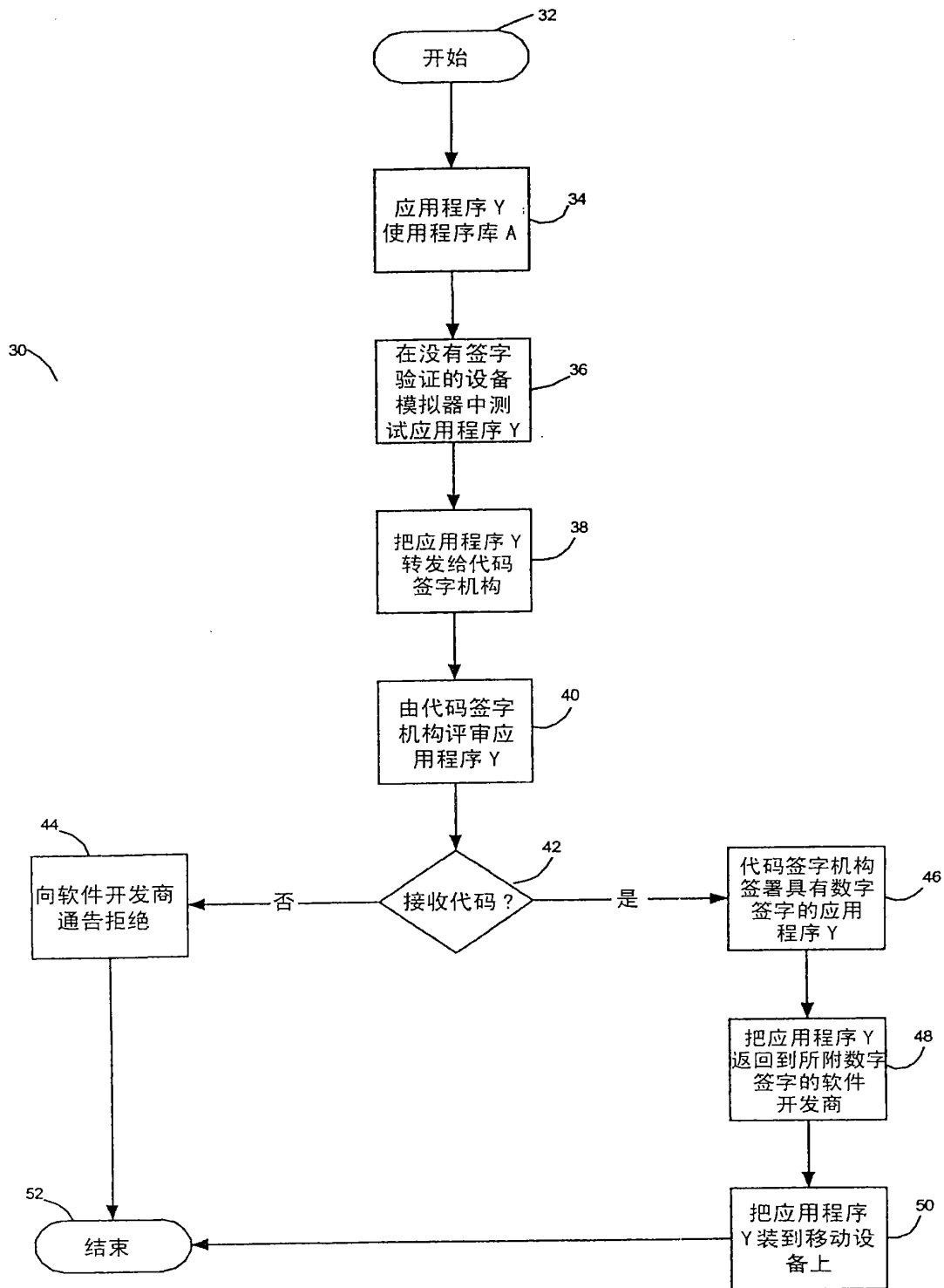


图 2

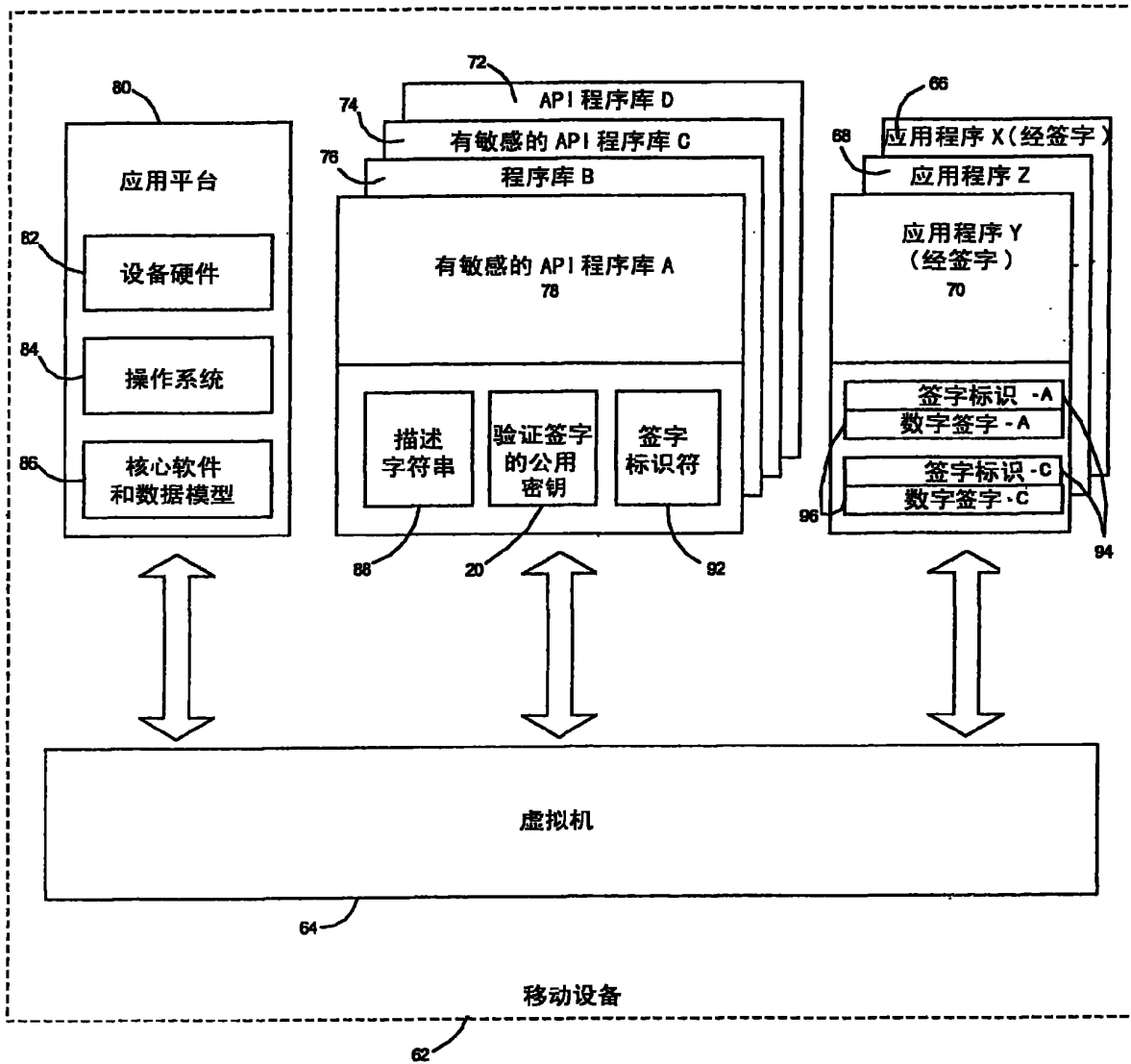


图 3

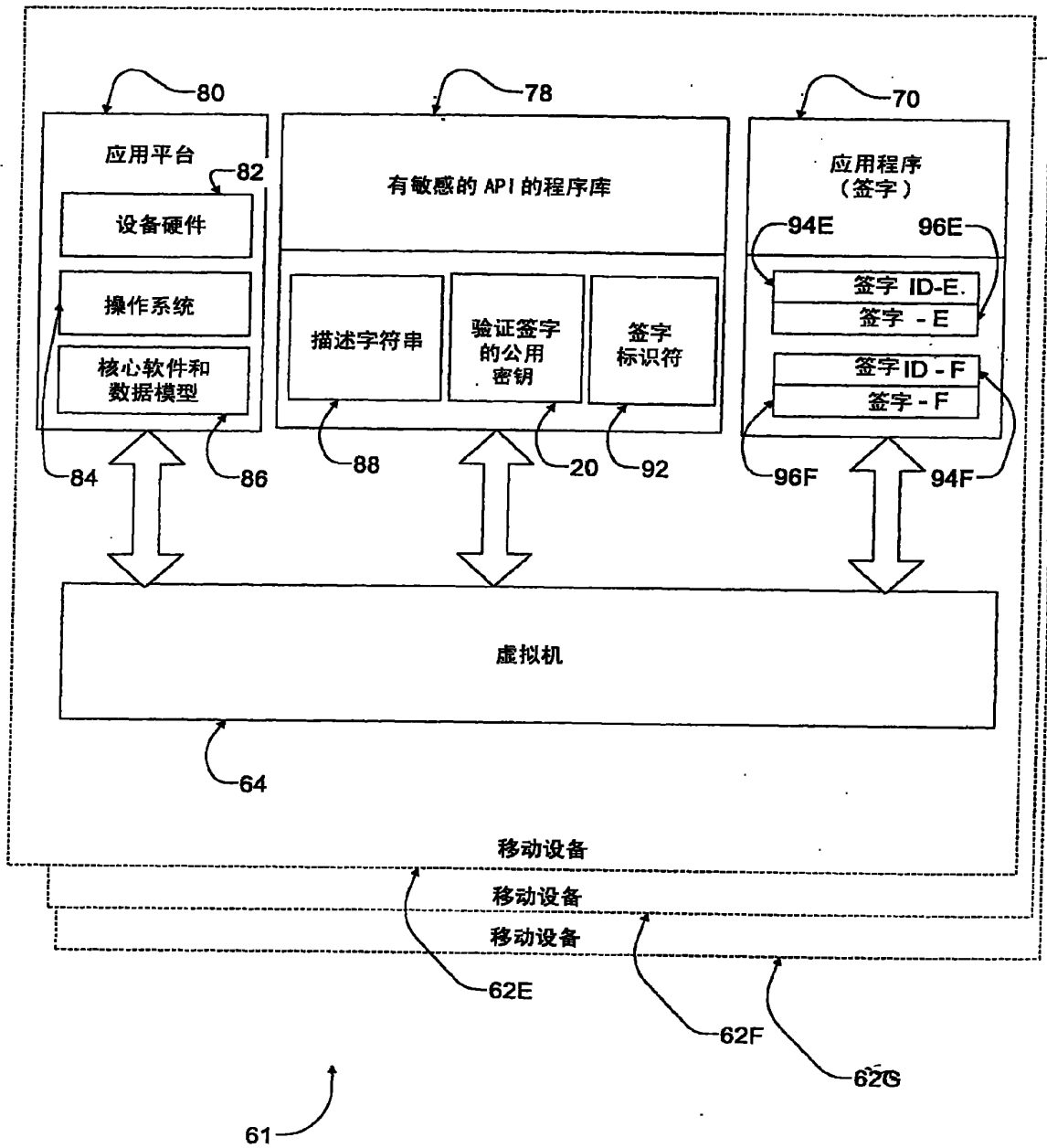


图 3A

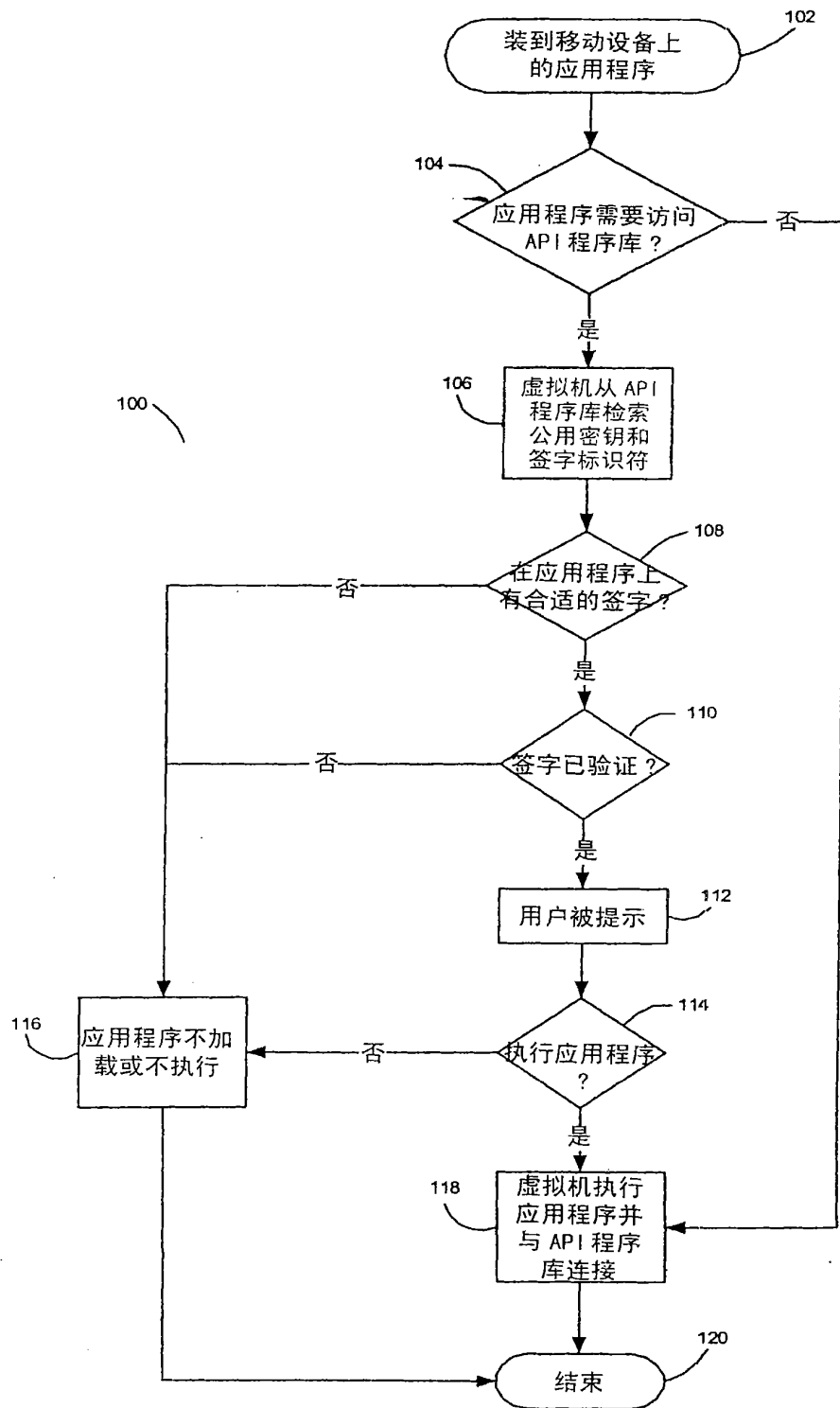


图 4

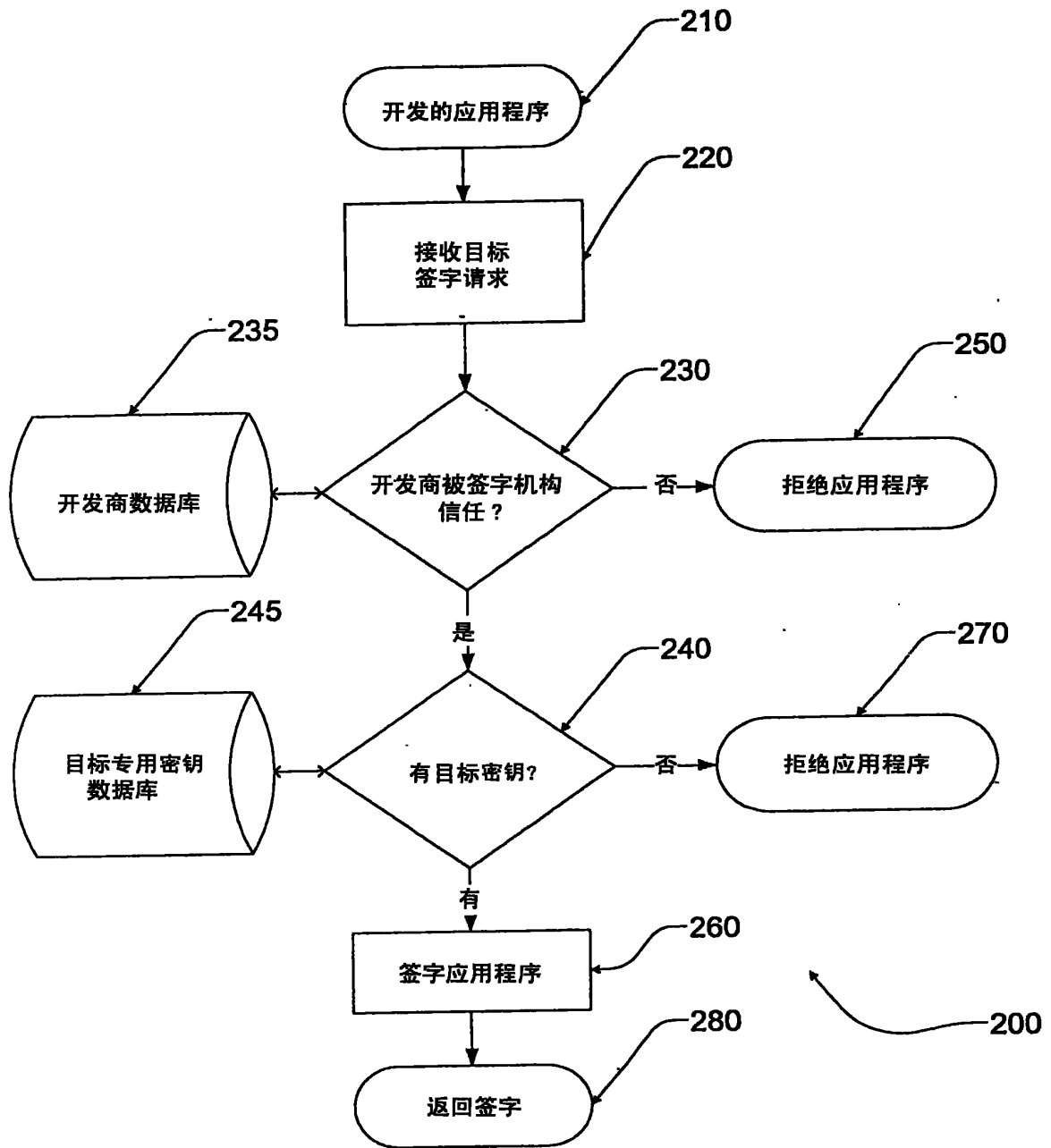


图 5

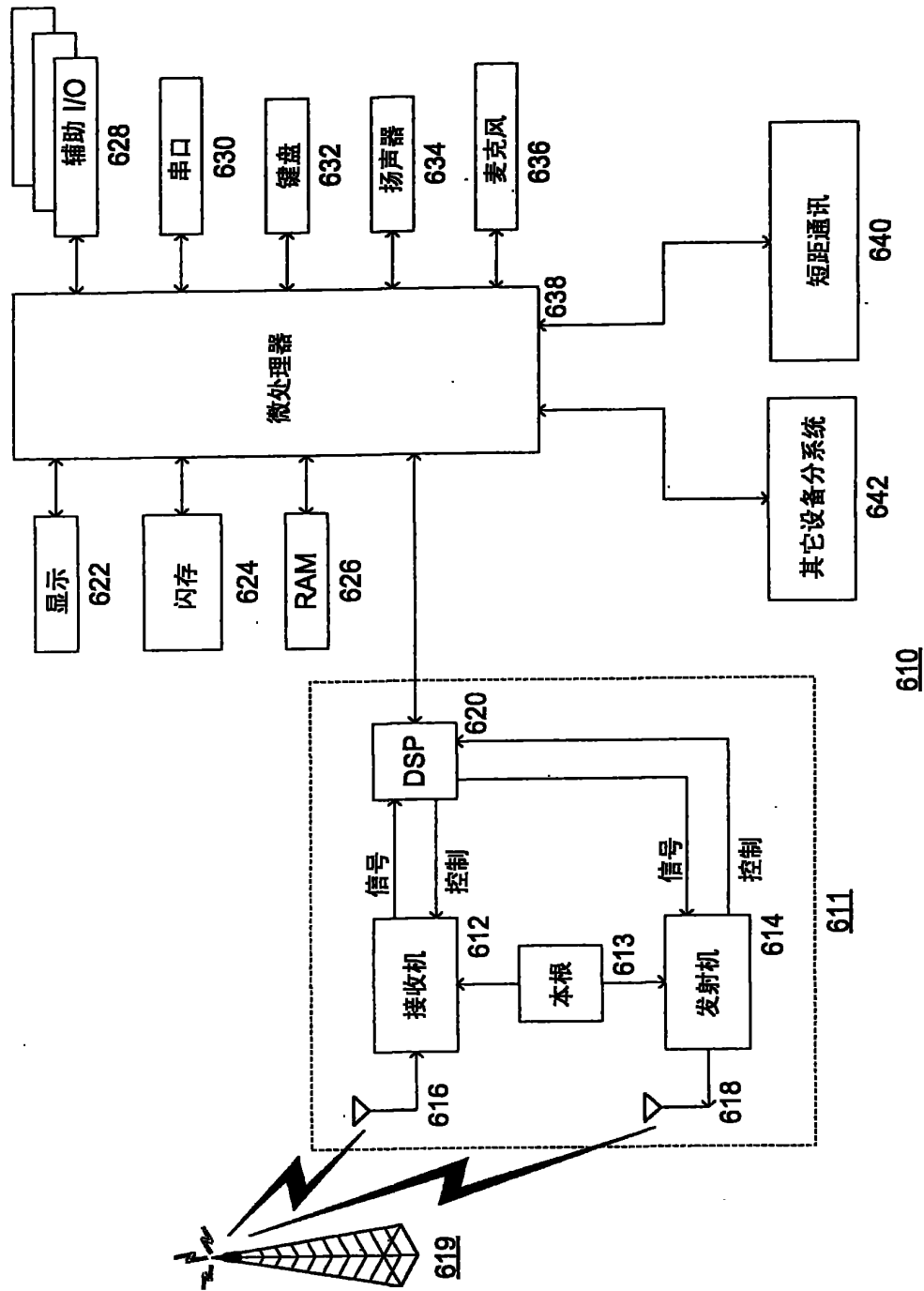


图 6

PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINARY EXAMINING AUTHORITY

PCT

To:
PATHIYAL, Krishna K.
RESEARCH IN MOTION LIMITED
295 Phillip Street
Waterloo, Ontario N2L 3W8
ETATS-UNIS D'AMERIQUE

WRITTEN OPINION

(PCT Rule 66)

Date of mailing (day/month/year) 28/05/2002	
Applicant's or agent's file reference PWO-0445	REPLY DUE within 1 / 00 months/days from the above date of mailing
International application No. PCT/CA 01/ 01344	International filing date (day/month/year) 20/09/2001
Priority date (day/month/year) 21/09/2000	
International Patent Classification (IPC) or both national classification and IPC G06F1/00	
Applicant RESEARCH IN MOTION LIMITED et al.	

1. This written opinion is the first drawn up by this International Preliminary Examining Authority.
2. This opinion contains indications relating to the following items:
 - I Basis of the opinion
 - II Priority
 - III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
 - IV Lack of unity of invention
 - V Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
 - VI Certain documents cited
 - VII Certain defects in the international application
 - VIII Certain observations on the international application
3. The applicant is hereby **invited to reply** to this opinion.

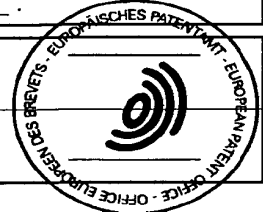
When? See the time limit indicated above. The applicant may, before the expiration of that time limit, request this Authority to grant an extension, see Rule 66.2(d).

How? By submitting a written reply, accompanied, where appropriate, by amendments, according to Rule 66.3. For the form and the language of the amendments, see Rules 66.8 and 66.9.

Also For an additional opportunity to submit amendments, see Rule 66.4.
For the examiner's obligation to consider amendments and/or arguments, see Rule 66.4bis.
For an informal communication with the examiner, see Rule 66.6.

If no reply is filed, the international preliminary examination report will be established on the basis of this opinion.
4. The final date by which the international preliminary examination report must be established according to Rule 69.2 is: 21/01/2003

Name and mailing address of the IPEA/ European Patent Office D-80298 Munich Tel. (+49-89) 2399-0, Tx: 523656 epmu d Fax: (+49-89) 2399-4465	Authorized officer Examiner Formalities officer (incl. extension of time limits) Tel. (+49-89) 2399 2828
---	--



I. Basis of the opinion

1. The basis of this written opinion is the application as originally filed.

V. Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability

1. In light of the documents cited in the international search report, it is considered that the invention as defined in at least some of the claims does not appear to meet the criteria mentioned in Article 33(1) PCT, i.e. does not appear to be novel and/or to involve an inventive step (see international search report, in particular the documents cited X and/or Y and corresponding claims references).
2. If amendments are filed, the applicant should comply with the requirements of Rule 66.8 PCT and indicate the basis of the amendments in the documents of the application as originally filed (Article 34 (2) (b) PCT) otherwise these amendments may not be taken into consideration for the establishment of the international preliminary examination report. The attention of the applicant is drawn to the fact that if the application contains an unnecessary plurality of independent claims, no examination of any of the claims will be carried out.

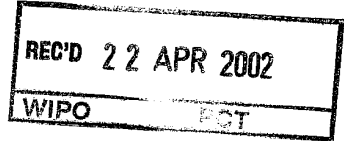
NB: Should the applicant decide to request detailed substantive examination, then an international preliminary examination report will normally be established directly. Exceptionally the examiner may draw up a second written opinion, should this be explicitly requested.

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)



Applicant's or agent's file reference PCA-0445	FOR FURTHER ACTION see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. PCT/CA 01/ 01344	International filing date (day/month/year) 20/09/2001	(Earliest) Priority Date (day/month/year) 21/09/2000
Applicant RESEARCH IN MOTION LIMITED		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 3 sheets.
 It is also accompanied by a copy of each prior art document cited in this report.

1. Basis of the report

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

- contained in the international application in written form.
- filed together with the international application in computer readable form.
- furnished subsequently to this Authority in written form.
- furnished subsequently to this Authority in computer readable form.
- the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
- the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. **Certain claims were found unsearchable** (See Box I).

3. **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

- the text is approved as submitted by the applicant.
- the text has been established by this Authority to read as follows:

SOFTWARE CODE SIGNING SYSTEM AND METHOD

5. With regard to the **abstract**,

- the text is approved as submitted by the applicant.
- the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

- as suggested by the applicant. 2
- because the applicant failed to suggest a figure. None of the figures.
- because this figure better characterizes the invention.

INTERNATIONAL SEARCH REPORT

national Application No
PCT/CA 01/01344

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 99 05600 A (APPLE COMPUTER) 4 February 1999 (1999-02-04) abstract; figures 5,6,9 page 6, line 1 - line 15 page 19, line 4 - line 14 page 20, line 19 -page 21, line 4 page 24, line 6 - line 23 page 25, line 23 - line 26	1, 2, 6, 7, 12-15, 21, 26, 27, 29, 32
Y	--- -/--	11, 18, 19, 26, 31, 38-56

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

° Special categories of cited documents :

<p>*A* document defining the general state of the art which is not considered to be of particular relevance</p> <p>*E* earlier document but published on or after the international filing date</p> <p>*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>*O* document referring to an oral disclosure, use, exhibition or other means</p> <p>*P* document published prior to the international filing date but later than the priority date claimed</p>	<p>*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>*&* document member of the same patent family</p>
--	--

Date of the actual completion of the international search 12 April 2002	Date of mailing of the international search report 22/04/2002
---	---

Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Powell, D
--	--

INTERNATIONAL SEARCH REPORT

International Application No
PCT/CA 01/01344

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 930 793 A (TEXAS INSTRUMENTS INC) 21 July 1999 (1999-07-21) abstract; figure 6 page 15, line 54 -page 16, line 5 page 16, line 32 - line 44	1,3-6, 8-10,20, 22-24, 28-33, 36,37
Y	-----	34,35
P,Y	US 6 157 721 A (SIBERT W OLIN ET AL) 5 December 2000 (2000-12-05) abstract; figures 2,3,5,8,14 column 2, line 27 - line 65 column 11, line 7 - line 19 column 15, line 23 - line 41	11,18, 19,26, 31,34, 35,38-56
Y	& AU 36815 97 A (INTERTRUST TECHNOLOGIES CORP) 19 February 1998 (1998-02-19) -----	
A	US 5 978 484 A (APPERSON NORMAN ET AL) 2 November 1999 (1999-11-02) abstract; figure 5 column 2, line 41 - line 60 column 3, line 44 - line 57 column 8, line 17 - line 25 -----	11,18, 19,31, 34,35

INTERNATIONAL SEARCH REPORT

Information on patent family members

national Application No
PCT/CA 01/01344

Patent document cited in search report	A	Publication date	Patent family member(s)	Publication date
WO 9905600	A	04-02-1999	US 6188995 B1	13-02-2001
			EP 1023664 A2	02-08-2000
			WO 9905600 A2	04-02-1999
EP 0930793	A	21-07-1999	CN 1249643 A	05-04-2000
			EP 0930793 A1	21-07-1999
			JP 11312152 A	09-11-1999
US 6157721	A	05-12-2000	AU 3205797 A	05-12-1997
			AU 3681597 A	19-02-1998
			CN 1225739 A	11-08-1999
			EP 0898777 A2	03-03-1999
			JP 2001501763 T	06-02-2001
			WO 9743761 A2	20-11-1997
			US 6292569 B1	18-09-2001
			US 2002023214 A1	21-02-2002
US 5978484	A	02-11-1999	NONE	

PATENT COOPERATION TREATY

From the
INTERNATIONAL PRELIMINARY EXAMINING AUTHORITY

To:

PATHIYAL, Krishna K.
RESEARCH IN MOTION LIMITED
295 Phillip Street
Waterloo, Ontario N2L 3W8
CANADA

PCT

NOTIFICATION OF TRANSMITTAL OF
THE INTERNATIONAL PRELIMINARY
EXAMINATION REPORT
(PCT Rule 71.1)

Date of mailing (day/month/year)	15.11.2002
-------------------------------------	------------

Applicant's or agent's file reference PWO-0445	IMPORTANT NOTIFICATION
---	-------------------------------

International application No. PCT/CA01/01344	International filing date (day/month/year) 20/09/2001	Priority date (day/month/year) 21/09/2000
---	--	--

Applicant RESEARCH IN MOTION LIMITED et al.
--


1. The applicant is hereby notified that this International Preliminary Examining Authority transmits herewith the international preliminary examination report and its annexes, if any, established on the international application.
2. A copy of the report and its annexes, if any, is being transmitted to the International Bureau for communication to all the elected Offices.
3. Where required by any of the elected Offices, the International Bureau will prepare an English translation of the report (but not of any annexes) and will transmit such translation to those Offices.
4. **REMINDER**

The applicant must enter the national phase before each elected Office by performing certain acts (filing translations and paying national fees) within 30 months from the priority date (or later in some Offices) (Article 39(1)) (see also the reminder sent by the International Bureau with Form PCT/IB/301).

Where a translation of the international application must be furnished to an elected Office, that translation must contain a translation of any annexes to the international preliminary examination report. It is the applicant's responsibility to prepare and furnish such translation directly to each elected Office concerned.

For further details on the applicable time limits and requirements of the elected Offices, see Volume II of the PCT Applicant's Guide.

For the purpose of deciding whether the claimed invention is patentable or not, the elected Offices may apply criteria additional to or different from the criteria on which the international preliminary examination report is based (see Articles 27(5), 33(5)). Additional criteria may include e.g. exemptions from patentability and the requirements of enabling disclosure and of clarity and support of claims.

Name and mailing address of the IPEA/  European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Authorized officer Koski, P Tel. +49 89 2399-2709
--	---


	
--	---

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference PWO-0445		FOR FURTHER ACTION	See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)
International application No. PCT/CA01/01344	International filing date (day/month/year) 20/09/2001	Priority date (day/month/year) 21/09/2000	
International Patent Classification (IPC) or national classification and IPC G06F1/00			
Applicant RESEARCH IN MOTION LIMITED et al.			
<p>1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.</p> <p>2. This REPORT consists of a total of 4 sheets, including this cover sheet.</p> <p><input type="checkbox"/> This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).</p> <p>These annexes consist of a total of sheets.</p>			
<p>3. This report contains indications relating to the following items:</p> <ul style="list-style-type: none"> I <input checked="" type="checkbox"/> Basis of the report II <input type="checkbox"/> Priority III <input checked="" type="checkbox"/> Non-establishment of opinion with regard to novelty, inventive step and industrial applicability IV <input type="checkbox"/> Lack of unity of invention V <input type="checkbox"/> Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement VI <input type="checkbox"/> Certain documents cited VII <input type="checkbox"/> Certain defects in the international application VIII <input type="checkbox"/> Certain observations on the international application 			
Date of submission of the demand 18/04/2002		Date of completion of this report 15.11.2002	
Name and mailing address of the international preliminary examining authority:  European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465		Authorized officer Kerschbaumer, J Telephone No. +49 89 2399 2999	



Form PCT/IPEA/409 (cover sheet) (January 1994)

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/CA01/01344

I. Basis of the report

1. With regard to the **elements** of the international application (*Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17)*):

Description, pages:

1-28 as originally filed

Claims, No.:

1-109 as received on 28/06/2002 with letter of 28/06/2002

Drawings, sheets:

1/7-7/7 as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

These elements were available or furnished to this Authority in the following language: , which is:

- the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
- the language of publication of the international application (under Rule 48.3(b)).
- the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

- contained in the international application in written form.
- filed together with the international application in computer readable form.
- furnished subsequently to this Authority in written form.
- furnished subsequently to this Authority in computer readable form.
- The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
- The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

- the description, pages:
- the claims, Nos.:

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/CA01/01344

the drawings, sheets:

5. This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

1. The questions whether the claimed invention appears to be novel, to involve an inventive step (to be non-obvious), or to be industrially applicable have not been examined in respect of:

the entire international application.

claims Nos. .

because:

the said international application, or the said claims Nos. relate to the following subject matter which does not require an international preliminary examination (*specify*):

the description, claims or drawings (*indicate particular elements below*) or said claims Nos. are so unclear that no meaningful opinion could be formed (*specify*):
see separate sheet

the claims, or said claims Nos. are so inadequately supported by the description that no meaningful opinion could be formed.

no international search report has been established for the said claims Nos. .

2. A meaningful international preliminary examination cannot be carried out due to the failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions:

the written form has not been furnished or does not comply with the standard.

the computer readable form has not been furnished or does not comply with the standard.

Re Item III

Although system claims 1, 6, 56, 77 and method claims 27, 36, 43, 47, 68, 87, 104 have been drafted as separate independent claims, they appear to relate effectively to the same subject-matter and to differ from each other only with regard to the definition of the subject-matter for which protection is sought or in respect of the terminology used for the features of that subject-matter. The aforementioned claims therefore lack conciseness. Moreover, lack of clarity of the claims as a whole arises, since the plurality of independent claims makes it impossible to determine the matter for which protection is sought, and places an undue burden on others seeking to establish the extent of the protection.

Hence, system claims 1, 6, 56, 77 and method claims 27, 36, 43, 47, 68, 87, 104 do not meet the requirements of Article 6 PCT.

We claim:

1. 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:
 - 5 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
 - 10 signature identification.
2. The code signing system of claim 1, wherein the virtual machine denies the software application access to the API if the digital signature is not authenticated.
- 15 3. The code signing system of claim 1, wherein the virtual machine purges the software application if the digital signature is not authenticated.
4. The code signing system of claim 1, wherein the code signing system is installed on a mobile device.
- 20 5. The code signing system of claim 1, wherein the digital signature is generated by a code signing authority.
6. 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
 - 30 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,

5 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.

7. The code signing system of claim 6, wherein the plurality of APIs are included in an API library.

10 8. The code signing system of claim 6, 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.

15 9. The code signing system of claim 8, 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
20 of the plurality of software applications.

10. The code signing system of claim 6, wherein the resource on the application platform comprises a wireless communication system.

25 11. The code signing system of claim 6, wherein the resource on the application platform comprises a cryptographic module which implements cryptographic algorithms.

12. The code signing system of claim 6, wherein the resource on the application platform
30 comprises a data store.

13. The code signing system of claim 6, wherein the resource on the application platform comprises a user interface (UI).

14. The code signing system of claim 1, further comprising:

5 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.

15. The code signing system of claim 14, wherein at least one of the plurality of API
10 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;

15 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
20 library.

16. The code signing system of claim 1, 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.

25

17. The code signing system of claim 16, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application; and

30 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.

18. The code signing system of claim 4, wherein the API further comprises:

5 a description string that is displayed by the mobile device when the software application attempts to access the API.

19. The code signing system of claim 1, wherein the application platform comprises an operating system.

10

20. The code signing system of claim 1, wherein the application platform comprises one or more core functions of a mobile device.

21. The code signing system of claim 1, wherein the application platform comprises hardware on a mobile device.

15

22. The code signing system of claim 23, wherein the hardware comprises a subscriber identity module (SIM) card.

23. The code signing system of claim 1, wherein the software application is a Java application for a mobile device.

20

24. The code signing system of claim 1, wherein the API interfaces with a cryptographic routine on the application platform.

25

25. The code signing system of claim 1, wherein the API interfaces with a proprietary data model on the application platform.

26. The code signing system of claim 1, wherein the virtual machine is a Java virtual machine installed on a mobile device.

30

27. 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. .

10

28. The method of claim 27, comprising the additional step of:

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

29. The method of claim 27, wherein the digital signature and the signature identification are generated by a code signing authority.

30. The method of claim 27, 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.

31. The method of claim 30, comprising the additional step of:

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

32. The method of claim 30, 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

5 comparing the generated hash with the recovered hash.

33. The method of claim 32, wherein the digital signature is generated by calculating a hash of the software application and applying the private signature key.

10 34. The method of claim 27, 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.

35. The method of claim 34, comprising the additional step of:

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

36. 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,
20 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;;

25 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.

37. The method of claim 36, wherein the step of determining whether the software application satisfies at least one criterion is performed by a code signing authority.

38. The method of claim 36, wherein the step of appending the digital signature and the
5 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.

10

39. The method of claim 38, wherein the hash of the software application is calculated using the Secure Hash Algorithm (SHA1).

40. The method of claim 38, wherein the step of verifying the authenticity of the digital
15 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;

20 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.

41. The method of claim 40, comprising the further step of denying the software application
25 access to the API where the digital signature is not authenticated..

42. The method of claim 40, wherein the signature key is a private signature key and the corresponding signature key is a public signature key.

43. 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;

5 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,;

10 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

15 corresponds with the signature identifier.

44. The method of claim 43, wherein the step of generating the digital signature is performed by a code signing authority.

20 45. The method of claim 43, wherein the step of generating the digital signature is performed by applying a signature key to the hash of the software application.

46. The method of claim 45, wherein the mobile device verifies the authenticity of the digital signature by performing the additional steps of:

25 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..

5

47. 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)

10

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 .

15

48. The method of claim 47, wherein the digital signature and signature identification are associated with a type of mobile device.

49. The method of claim 47, comprising the additional step of:

20

purging the software application from the mobile device where the software application does not include an authentic digital signature. .

50. The method of claim 47, wherein:

25

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.

51. The method of claim 50, wherein each of the plurality of digital signatures and associated signature identifications are generated by a respective corresponding code signing authority.
52. The method of claim 47, wherein the step of determining whether the software application includes an authentic digital signature comprises the additional steps of:
5 verifying the authenticity of the digital signature where the signature identification corresponds with respective ones of the at least one signature identifier.
53. The method of claim 51, 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.
10
54. The method of claim 47, wherein the step of authenticating the digital signature where the signature identification corresponds with the signature identifier comprises the steps of:
15 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:
20 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.
- 25 55. The method of claim 47, 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
30 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;

- 5 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 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

- 10 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; and

- 15 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.

- 20 56. A code signing system for controlling access to application programming interfaces (APIs) having signature identifiers by software applications, the code signing system comprising:

- 25 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 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; and

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.

57. The code signing system of claim 56, wherein a virtual machine comprises the verification system and the control system.
58. The code signing system of claim 57, wherein the virtual machine is a Java virtual machine installed on a mobile device.
59. The code signing system of claim 56, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.
60. The code signing system of claim 56, wherein the code signing system is installed on a mobile device and the software application is a Java application for a mobile device.
61. The code signing system of claim 56, wherein the digital signature and the signature identification of the software application are generated by a code signing authority.
62. The code signing system of claim 56, 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).
63. The code signing system of claim 56, 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.
64. The code signing system of claim 63, 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.

65. The code signing system of claim 56, 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.
- 5
66. The code signing system of claim 56, 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.
- 10 67. The code signing system of claim 56, wherein verification of a global digital signature
provided by the software application is required for accessing any of the APIs.
68. A method of controlling access to application programming interfaces (APIs) having
signature identifiers by software applications, the method comprising:
- 15 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; and
allowing access to at least one of the APIs where the digital signature provided by the
20 software application is authenticated.
69. The method of claim 68, wherein one digital signature and one signature identification
are provided by the software application access a library of at least one of the APIs.
- 25 70. The method of claim 68, wherein the digital signature and the signature identification of
the software application are generated by a code signing authority.
71. The method of claim 68, 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
30 interface (UI).

72. The method of claim 68, 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.

5

73. The method of claim 72, wherein:

the digital signature is generated by applying the private signature key to a hash of the software application under the signature scheme; and

10 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.

74. The method of claim 68, wherein at least one of the APIs further comprises:

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

75. The method of claim 68, 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.

20 76. The method of claim 68, wherein verification of a global digital signature provided by the software application is required for accessing any of the APIs

77. 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
25 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
30 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

5 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; and

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.

10

78. The management system of claim 77, wherein a virtual machine comprises the verification system and the control system.

79. The management system of claim 78, wherein the virtual machine is a Java virtual
15 machine and the software applications are Java applications.

80. The management system of claim 77, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

20 81. The management system of claim 77, 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).

25 82. The management system of claim 77, 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.

83. The management system of claim 82, wherein:
the digital signature is generated by applying the private signature key to a hash of the
30 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.

5

84. The management system of claim 77, 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.

10

85. The management system of claim 77, 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.

15

86. The management system of claim 77, 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.

20

87. 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

25

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; wherein each mobile device of the subset of the plurality of mobile devices comprises

30

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; and

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.

5 88. The method of claim 87, wherein a virtual machine comprises the verification system and the control system.

89. The method of claim 88, wherein the virtual machine is a Java virtual machine and the software applications are Java applications.

10

90. The method of claim 87, wherein the control system requires one digital signature and one signature identification for each library of at least one of the APIs.

15

91. The method of claim 87, 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).

20

92. The method of claim 87, 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.

25

93. The method of claim 92, 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.

30

94. The method of claim 87, 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.
- 5 95. The method of claim 87, 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.
96. A mobile device for a subset of a plurality of mobile devices, the mobile device
comprising:
10 an application platform having application programming interfaces (APIs);
a verification system for authenticating digital signatures and signature identifications
provided by 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 a digital signature provided by the software application is authenticated by the verification
15 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
20 comprise those signature identifications that are substantially only authorized to allow access on
the subset of the plurality of mobile devices.
97. The mobile device of claim 96, wherein a virtual machine comprises the verification
system and the control system.
25
98. The mobile device of claim 97, wherein the virtual machine is a Java virtual machine and
the software application is a Java application.
99. The mobile device of claim 96, wherein the control system requires one digital signature
30 and one signature identification for each library of at least one of the APIs.

100. The mobile device of claim 96, 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

101. The mobile device of claim 96, 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.

10 102. The mobile device of claim 101, 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.

15

103. The mobile device of claim 96, 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.

20

104. 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:

25 receiving digital signatures and signature identifications from software applications that require to access the APIs

authenticating the digital signatures and the signature identifications; and

allowing a software application to access at least one of the APIs where a digital signature provided by 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 subset of the plurality of mobile devices.

105. The method of claim 104, wherein one digital signature and one signature identification is required for accessing each library of at least one of the APIs.

106. The method of claim 104, 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).

107. The method of claim 104, 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.

108. The method of claim 107, 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.

109. The method of claim 104, 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.

Electronic Acknowledgement Receipt

EFS ID:	11983225
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:	89951
Filer:	Kendrick Lo./Shelley Cotgrave
Filer Authorized By:	Kendrick Lo.
Attorney Docket Number:	13210-1465/KL
Receipt Date:	02-FEB-2012
Filing Date:	20-MAR-2003
Time Stamp:	13:00:37
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	Transmittal Letter	10289_US_PCT_transmittal.pdf	151629 82affa1dfa324f5f3c061b6d18dc28ebf29d9d82	no	1

Warnings:

Information:

2	Transmittal Letter	10289_US_PCT_letter.pdf	96811 552757b08bc24fe122ef27b86df8ea0b5580321c	no	1
Warnings:					
Information:					
3	Information Disclosure Statement (IDS) Form (SB08)	10289_US_PCT_IDS.pdf	613458 e755537c0319ca4fd993709236a6ece286f5eb67	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.					
4	Foreign Reference	WO02_25409.pdf	886085 5b307514a9212cf5c206dd076d0a9097a1f103bd	no	50
Warnings:					
Information:					
5	Foreign Reference	CN101694687.pdf	1578896 29ba195541d97672123e62f4b5d3b3b88daabac4	no	26
Warnings:					
Information:					
6	Non Patent Literature	CA_abandonment15Nov11.pdf	157325 a3510a99de0fa70506a3e53229326fd30c085ed	no	1
Warnings:					
Information:					
7	Non Patent Literature	CA_NOA27Sep10.pdf	153583 1a76dbfab29ecaae4ce93b447893f896e2496dc7	no	1
Warnings:					
Information:					
8	Non Patent Literature	CA_OA_04Mar09.pdf	266649 d9633e3e8563685da2fcaa735713ed0e6e4ff505	no	4
Warnings:					
Information:					
9	Non Patent Literature	CA_OA_13Mar08.pdf	163218 9bf1ecfed42e4d6e21e349cc80c1653d986df0c35	no	4
Warnings:					
Information:					

10	Non Patent Literature	Written_Opinion_28May02.pdf	88526 03fdb6889c1b19bc017ea35f12742ec2ec61d906	no	2
Warnings:					
Information:					
11	Non Patent Literature	ISR_22Apr02.pdf	153103 bfd89ab484cd44ea6b255898da25a875aa327973	no	4
Warnings:					
Information:					
12	Non Patent Literature	EP1320795_Prel_Exam15Nov02.pdf	990238 2768a9b7f2b19058efe6d0987779adb120bec9f	no	25
Warnings:					
Information:					
13	Non Patent Literature	EP1320795_intent_to_grant_06May05.pdf	164431 9d8ff9878eafc9d2d0a5dfad944a22075d052767	no	5
Warnings:					
Information:					
14	Non Patent Literature	EP_1320795_opposition_notice21Aug06.pdf	31230 407c1fdb868662c4999c60bb2bdf78962f841810	no	1
Warnings:					
Information:					
15	Non Patent Literature	EP1320795_observations07May07.pdf	1094692 b9dafd7b93cd2d99a6446c2fa20f8995e013b4ba	no	29
Warnings:					
Information:					
16	Non Patent Literature	HANDBUCH.pdf	716015 3c63988b583b12b3627c702d59fb1aec55b6cdba	no	11
Warnings:					
Information:					
17	Non Patent Literature	EP1320795_Summons20Mar08.pdf	46720 8fa719484fd38ec22507109ddb10358afcb95917	no	1
Warnings:					
Information:					
18	Non Patent Literature	EP_minutes_22Dec08.pdf	211905 64f029997d007e29f0150758b143c7e212814f3	no	6
Warnings:					
Information:					

19	Non Patent Literature	EP1320795_Interlocutory_decision22Dec08.pdf	340109 94fefba6d19cc9db348c87562f152629c195eae9	no	12
Warnings:					
Information:					
20	Non Patent Literature	CN1541350_1st_OA.pdf	921245 a21d0ad421c86eb9dd138146e4e2ad48689aa5d4	no	10
Warnings:					
Information:					
21	Non Patent Literature	CN1541350_2nd_OA.pdf	262385 b34fad6ebaa6a1d31f5db89bb911ca8cb75b35e	no	7
Warnings:					
Information:					
22	Non Patent Literature	CN1541350_rejection.pdf	95103 ff9c085720734cd12df990c7f3c25c8c6db89b4	no	2
Warnings:					
Information:					
23	Non Patent Literature	CN1541350_reexamination.pdf	1037848 f43140ec55855a2164ac68d7b322126390591c03	no	10
Warnings:					
Information:					
24	Non Patent Literature	CN1541350_3rd_OA.pdf	207807 b29b015446751d69daefe91ba9637c468692e200	no	5
Warnings:					
Information:					
25	Non Patent Literature	CN1541350_certificate.pdf	37036 31390b89ee87d2c8902be19243d3c6ea06d7b9af	no	1
Warnings:					
Information:					
26	Non Patent Literature	EP1626324_loss_of_rights.pdf	46657 c5ca052193b51be9fbac930e4dc62051f7f04c6	no	1
Warnings:					
Information:					
27	Non Patent Literature	EP1626324_intent_to_grant.pdf	163981 c6dd83cba2e1e9851c222c7f4d6d5f1e81f28a67	no	4
Warnings:					
Information:					

28	Non Patent Literature	EP1626324_search_report15May09.pdf	94815	no	4
			364df4191c1382dbf46e1627eb52fb5d4257b44c		
Warnings:					
Information:					
29	Non Patent Literature	EP1626325_intent_to_grant.pdf	143613	no	4
			1a9d60c78ad4f8640c2c5103ead89a399ed0e2a2		
Warnings:					
Information:					
30	Non Patent Literature	EP1626326_search_report15May09.pdf	94920	no	4
			bbb63dd036a113f1f5421c61dc3278a551b73dda		
Warnings:					
Information:					
31	Non Patent Literature	EP1626326_intent_to_grant.pdf	143606	no	4
			1461f9104a81e3b77054d67d7bcc1f543dd174ad		
Warnings:					
Information:					
32	Non Patent Literature	EP2284644_search_report30Dec10.pdf	103056	no	4
			3354993eb6ecc69c59c37b902ebb242c64449be2		
Warnings:					
Information:					
33	Non Patent Literature	EP2278429_search_report_21Dec10.pdf	102918	no	4
			7cd816f5202b2620656746b2ace98c723e9b2bce		
Warnings:					
Information:					
34	Non Patent Literature	XP002269400.pdf	3890237	no	36
			8257abb206c68ff44233f270b99f6f48cced674		
Warnings:					
Information:					
Total Files Size (in bytes):			15249850		

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.

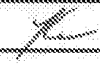
New International Application Filed with the USPTO as a Receiving Office

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.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>	Application Number	10/381,219
	Filing Date	March 20, 2003
	First Named Inventor	YACH, David P.
	Art Unit	2431
	Examiner Name	AVERY, Jeremiah L.
Total Number of Pages in This Submission	Attorney Docket Number	13210-1465/KL

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> After Allowance Communication to TC
<input type="checkbox"/> Fee Attached	<input type="checkbox"/> Licensing-related Papers	<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
<input type="checkbox"/> Amendment/Reply	<input type="checkbox"/> Petition	<input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
<input type="checkbox"/> After Final	<input type="checkbox"/> Petition to Convert to a Provisional Application	<input type="checkbox"/> Proprietary Information
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Power of Attorney, Revocation	<input type="checkbox"/> Status Letter
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Change of Correspondence Address	<input checked="" type="checkbox"/> Other Enclosure(s) (please identify below):
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Terminal Disclaimer	Copies of documents cited in IDS.
<input checked="" type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> Request for Refund	
<input type="checkbox"/> Certified Copy of Priority Document(s)	<input type="checkbox"/> CD, Number of CD(s) _____	
<input type="checkbox"/> Reply to Missing Parts/Incomplete Application	<input type="checkbox"/> Landscape Table on CD	
<input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	Remarks	

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT		
Firm Name	Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.	
Signature		
Printed name	Kendrick Lo	
Date	February 2, 2012	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

This collection of information is required by 37 CFR 1.5. 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 1.11 and 1.14. This collection is estimated to 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, 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Bereskin & Parr

INTELLECTUAL PROPERTY LAW

February 2, 2012

Kendrick Lo B.A.Sc. (Eng. Sci.), MBA, LL.B.
416 957 1685 klo@bereskinparr.com

Your Reference: 10/381,219
Our Reference: 13210-1465/KL

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA
22313-1450

Dear Sir:

Re: U.S. Patent Application No. 10/381,219
For: SOFTWARE CODE SIGNING SYSTEM AND METHOD
Filing Date: March 20, 2003
Applicants: YACH, David P. et al.

In accordance with 37 C.F.R. 1.56 and 1.97(b)(4), the Applicants hereby submit a Supplemental Information Disclosure Statement including (1) a listing, on PTO form SB/08a, of patents and other publications of which the Applicants are aware that may be considered material to patentability, and (2) a copy of the foreign patent documents and the non-patent literature documents.

Please note that we are resubmitting the reference cited at Cite No. 28 to correct a typographical error in the date of the document which was originally submitted in an IDS on November 11, 2011 at Cite No. 7.

The filing of this statement shall be not construed as an admission that the information cited in the attached statement is, or is considered to be, material to patentability (37 CFR 1.97(h)), nor as an admission that it constitutes prior art.

Please have the document recorded against the above-mentioned application.

Respectfully submitted,

BERESKIN & PARR LLP/S.E.N.C.R.L., s.r.l.

By 

Kendrick Lo
Reg. No. 54,948
Tel: (416) 364-7311

Encl.

Bereskin & Parr LLP
Scotia Plaza, 40 King Street West, 40th Floor, Toronto, Ontario, Canada M5H 3Y2
Tel: 416.364.7311 Fax: 416.361.1398 www.bereskinparr.com

TORONTO MISSISSAUGA WATERLOO MONTRÉAL



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/381,219	03/20/2003	David P Yach	555255012423

CONFIRMATION NO. 9761

POWER OF ATTORNEY NOTICE



89441
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

Date Mailed: 11/30/2011

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/11/2011.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/snguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/381,219	03/20/2003	David P Yach	555255012423

CONFIRMATION NO. 9761

POA ACCEPTANCE LETTER



89951
Bereskin and Parr LLP
S.E.N.C.R.L., s.r.l.
40 King Street West
40th Floor
Toronto, ON M5H 3Y2
CANADA

Date Mailed: 11/30/2011

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/11/2011.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/snguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



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

Table with 7 columns: APPLICATION NUMBER, FILING or 371(c) DATE, GRP ART UNIT, FIL FEE REC'D, ATTY.DOCKET.NO, TOT CLAIMS, IND CLAIMS. Row 1: 10/381,219, 03/20/2003, 2431, 4158, 555255012423, 109, 12

CONFIRMATION NO. 9761

CORRECTED FILING RECEIPT



0000000011996155

89441
Jones Day (RIM) - 2N
North Point
901 Lakeside Avenue
Cleveland, OH 44114

Date Mailed: 11/25/2011

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections

Applicant(s)

David P Yach, Waterloo, ON, CANADA;
Michael S Brown, Waterloo, ON, CANADA;
Herbert A Little, Waterloo, ON, CANADA;

Power of Attorney:

David Cochran--39142

Domestic Priority data as claimed by applicant

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 (You may be eligible to benefit from the Patent Prosecution Highway program at the USPTO. Please see http://www.uspto.gov for more information.)

If Required, Foreign Filing License Granted: 02/27/2004

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is US 10/381,219

Projected Publication Date: Not Applicable

Non-Publication Request: No

Early Publication Request: No

Title

Software code signing system and method

Preliminary Class

713

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER**Title 35, United States Code, Section 184****Title 37, Code of Federal Regulations, 5.11 & 5.15****GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as

set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:

Practitioners associated with the Customer Number: 89951

OR

Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:

The address associated with Customer Number: 89951

OR


<input type="checkbox"/> Firm or Individual Name			
Address			
City	State	Zip	
Country			
Telephone	Email		

Assignee Name and Address:
 Research In Motion Limited
 295 Phillip Street
 Waterloo, Ontario, CANADA N2L 3W8

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	9/10/09
Name	Brian Bidulka	Telephone	(519) 888-7465
Title	Chief Accounting Officer		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. 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 1.11 and 1.14. This collection is estimated to take 3 minutes 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.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Legal OK
 DW MP

RIM OK

Privacy Act Statement


The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

RIM OK

Under the paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a)		Docket Number (Optional) 13210-1465/KL	
Application Number 10/381,219		Filed March 20, 2003	
For SOFTWARE CODE SIGNING SYSTEM AND METHOD			
Art Unit 2431		Examiner AVERY, Jeremiah L.	
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):			
		<u>Fee</u>	<u>Small Entity Fee</u>
<input type="checkbox"/>	One month (37 CFR 1.17(a)(1))	\$150	\$75 \$ _____
<input type="checkbox"/>	Two months (37 CFR 1.17(a)(2))	\$560	\$280 \$ _____
<input checked="" type="checkbox"/>	Three months (37 CFR 1.17(a)(3))	\$1270	\$635 \$ <u>1270</u>
<input type="checkbox"/>	Four months (37 CFR 1.17(a)(4))	\$1980	\$990 \$ _____
<input type="checkbox"/>	Five months (37 CFR 1.17(a)(5))	\$2690	\$1345 \$ _____
<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27.			
<input type="checkbox"/> A check in the amount of the fee is enclosed.			
<input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.			
<input type="checkbox"/> The Director has already been authorized to charge fees in this application to a Deposit Account.			
<input checked="" type="checkbox"/> The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number <u>022095</u> .			
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 <input type="checkbox"/> applicant/inventor.			
<input type="checkbox"/> 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).			
<input checked="" type="checkbox"/> attorney or agent of record. Registration Number <u>54,948</u>			
<input type="checkbox"/> attorney or agent under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34 _____			
		November 11, 2011	
Signature		Date	
Kendrick Lo		416-364-7311	
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.			
<input checked="" type="checkbox"/> Total of <u>1</u> forms are submitted.			

This collection of information is required by 37 CFR 1.136(a). 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 1.11 and 1.14. This collection is estimated to take 6 minutes 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.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

**REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
 (Submitted Only via EFS-Web)**

Application Number	12016632	Filing Date	2008-01-18	Docket Number (if applicable)	13210-1465/KL	Art Unit	2617
First Named Inventor	KIRKUP, Michael G.			Examiner Name	LY, Nghi H.		

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 _____
- Enclosed
- Amendment/Reply
- Information Disclosure Statement (IDS)
- Affidavit(s)/ Declaration(s)
- Other _____

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 022095

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

- Patent Practitioner Signature
- Applicant Signature

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.

Signature of Registered U.S. Patent Practitioner			
Signature	/Kendrick Lo/	Date (YYYY-MM-DD)	2011-11-11
Name	Kendrick Lo	Registration Number	54948

This collection of information is required by 37 CFR 1.114. 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 1.11 and 1.14. This collection is estimated to take 12 minutes 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.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Appl. No. 10/381,219
Amendment dated November 11, 2011
Reply to office action of May 13, 2011

Appl. No : 10/381,219
Applicants : YACH, David et al.
Filed : March 20, 2003
Title : Software Code Signing System and Method

TC./A.U. : 2431
Examiner : AVERY, Jeremiah L

Docket No. : 13210-1465/KL (previously 555255012423)

November 11, 2011

Commissioner for Patents
P. O. Box 1450
Alexandria, Virginia 22313-1450

AMENDMENT

Sir:

In response to the office action of May 13, 2011, please amend the above-identified application as follows. A petition for a three-month extension of time and a request for continued examination accompany this response.

Amendments to the Claims are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks/Arguments begin on page 23 of this reply.

Amendments to the Claims

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1-165: (Cancelled without prejudice).

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 of the mobile device, the operations comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, 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 to which access is restricted~~accessible upon verification of a digital signature, wherein the sensitive API is ~~associated with a signature identifier and a public key;~~

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to [[a]] the sensitive API stored at the mobile device;

~~using an application execution manager to:~~

~~determine~~ determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device; 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, the mobile device using a use the public key associated with the sensitive API of the private key-~~

public key pair to verify authenticity of the digital signature of the signed software application; and

based upon verifying the authenticity of the digital signature at the mobile device, using the sensitive API to the mobile device allowing the signed software application access to the sensitive API to interact with the application platform.

167. (Currently Amended) 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 operations further comprise: preventing execution of the software application is not executed.

168. (Currently Amended) 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 operations further comprise: denying the software application is denied access to the sensitive API.

169. (Currently Amended) 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 operations further comprise: purging the software application from the mobile device.

170. (Currently Amended) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified authenticated, the operations further comprise: preventing execution of the software application requesting access to the sensitive API is not executed.

171. (Currently Amended) The mobile device of claim 166, wherein based upon a determination that the digital signature is not successfully verified authenticated, the operations further comprise: denying the software application requesting access to the sensitive API is denied access to the sensitive API.

172. (Currently Amended) 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 successfully verified ~~authenticated~~, ~~the application execution manager purges the~~ operations further comprise: purging the software application requesting access to the sensitive API from the mobile device.

173. (Currently Amended) 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 access the sensitive API ~~interact with the application platform.~~

174. (Currently Amended) The mobile device of claim 166, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

175. (Currently Amended) The mobile device of claim 166, wherein the digital signature is generated by applying ~~[[a]]~~ the private key to a first hash of the software application; and the digital signature is verified ~~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. (Currently Amended) The mobile device of claim 166, wherein the digital signature is generated by applying ~~[[a]]~~ the private key to a first abridged version of the software application; and the digital signature is verified ~~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. (Currently Amended) 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. (Currently Amended) 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. (Currently Amended) 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. (Currently Amended) The mobile device of claim 166, wherein the software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

190. (Currently Amended) The mobile device of claim 166, wherein the operations further comprise 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 on a mobile device, 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) at the mobile device, 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 to which access is restricted accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;~~

~~receiving, at the mobile device, an indication that a software application is requesting access to [[a]] the sensitive API stored at the mobile device;~~

~~using an application execution manager to:~~

~~determine determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device; 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 mobile device using a [[the]] public key associated with the sensitive API of the private key-public key pair to verify authenticity of the digital signature of the signed software application; and~~

~~based upon verifying the authenticity of the digital signature, using the sensitive API to the mobile device allowing the signed software application access to the sensitive API to interact with the application platform.~~

192. (Currently Amended) 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 operations further comprise: preventing execution of the software application is not executed.

193. (Currently Amended) 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 operations further comprise: denying the
software application ~~is denied~~ access to the sensitive API.

194. (Currently Amended) 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~~ operations further
comprise: purging the software application from the mobile device.

195. (Currently Amended) The system of claim 191, wherein based upon a
determination that the digital signature is not successfully verified ~~authenticated~~, the
operations further comprise: preventing execution of the software application
~~requesting access to the sensitive API is not executed.~~

196. (Currently Amended) The system of claim 191, wherein based upon a
determination that the digital signature is not successfully verified ~~authenticated~~, the
operations further comprise: denying the software application requesting access to the
~~sensitive API is denied~~ access to the sensitive API.

197. (Currently Amended) 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 successfully verified ~~authenticated~~, ~~the application execution manager purges the~~
operations further comprise: purging the software application requesting access to the
~~sensitive API from the mobile device.~~

198. (Currently Amended) 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 access the sensitive API ~~interact with~~
~~the application platform.~~

199. (Currently Amended) The system of claim 191, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

200. (Currently Amended) The system of claim 191, wherein the digital signature is generated by applying [[a]] the private key to a first hash of the software application; and the digital signature is verified ~~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. (Currently Amended) The system of claim 191, wherein the digital signature is generated by applying [[a]] the private key to a first abridged version of the software application; and the digital signature is verified ~~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. (Currently Amended) 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]] the sensitive API, and based upon a determination that the software application should be given access to [[a]] the sensitive API, the code signing authority accepts the software application and generates [[a]] the digital signature that is included with the software application.

203. (Currently Amended) 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. (Currently Amended) 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. (Currently Amended) 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. (Currently Amended) The system of claim 191, wherein the software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

215. (Currently Amended) The system of claim 191, wherein the operations further comprise associating the sensitive API with the public key includes obtaining the public key from a public key repository.

216. (Currently Amended) A non-transitory computer-readable storage medium encoded with instructions that when executed on one or more processors of a mobile device, cause the mobile device to ~~within a computer system~~ perform a method for controlling access to an application platform of the mobile device, the method comprising:

storing a plurality of application programming interfaces (APIs) at the mobile device, ~~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 to which access is restricted~~ accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;

receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to [[a]] the sensitive API stored at the mobile device;
~~using an application execution manager to:~~

determine determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device; 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 mobile device using the public key associated with the sensitive API of the private key-public key pair to verify authenticity of the digital signature of the signed software application; and~~

~~based upon verifying the authenticity of the digital signature at the mobile device, using the sensitive API to the mobile device allowing the signed software application access to the sensitive API to interact with the application platform.~~

217. (Currently Amended) 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 method further comprises: preventing execution of the software application is not executed.

218. (Currently Amended) 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 method further comprises: denying the software application is denied access to the sensitive API.

219. (Currently Amended) 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~~ method further comprises: purging the software application from the mobile device.

220. (Currently Amended) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified

~~authenticated, the method further comprises: preventing execution of the software application requesting access to the sensitive API is not executed.~~

221. (Currently Amended) The computer-readable storage medium of claim 216, wherein based upon a determination that the digital signature is not successfully verified ~~authenticated, the method further comprises: denying the software application requesting access to the sensitive API is denied access to the sensitive API.~~

222. (Currently Amended) 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 successfully verified ~~authenticated, the application execution manager purges the method further comprises: purging the software application requesting access to the sensitive API from the mobile device.~~

223. (Currently Amended) 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 access the sensitive API ~~interact with the application platform.~~

224. (Currently Amended) The computer-readable storage medium of claim 216, wherein at least some of the operations are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

225. (Currently Amended) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying ~~[[a]]~~ the private key to a first hash of the software application; and the digital signature is verified ~~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. (Currently Amended) The computer-readable storage medium of claim 216, wherein the digital signature is generated by applying ~~[[a]]~~ the private key to a first abridged version of the software application; and the digital signature is verified ~~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. (Currently Amended) 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. (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. (Currently Amended) 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. (Currently Amended) The computer-readable storage medium of claim 216, wherein the software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

240. (Currently Amended) The computer-readable storage medium of claim 216, wherein the method further comprises ~~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 of a mobile device, comprising:

~~storing a plurality of application programming interfaces (APIs) at the mobile device, 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 to which access is restricted accessible upon verification of a digital signature, wherein the sensitive API is associated with a signature identifier and a public key;~~

~~receiving, at the mobile device, an indication that a software application on the mobile device is requesting access to ~~[[a]]~~ the sensitive API stored at the mobile device;~~

~~using an application execution manager to:~~

~~determine determining, at the mobile device, whether the software application is signed, wherein a signed software application includes a digital signature generated using a private key of a private key-public key pair, wherein the private key is not accessible to the mobile device; 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 mobile device using a public key associated with the sensitive API of the private key-public key pair to verify authenticity of the digital signature of the signed software application; and~~

~~based upon verifying the authenticity of the digital signature at the mobile device, using the sensitive API to the mobile device allowing the signed software application access to the sensitive API to interact with the application platform.~~

242. (Currently Amended) 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 method further comprises: preventing execution of the software application is not executed.

243. (Currently Amended) 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 method further comprises: denying the software application ~~is denied~~ access to the sensitive API.

244. (Currently Amended) 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~~ method further comprises: purging the software application from the mobile device.

245. (Currently Amended) The method of claim 241, wherein based upon a determination that the digital signature is not successfully verified authenticated, the method further comprises: preventing execution of the software application ~~requesting access to the sensitive API is not executed~~.

246. (Currently Amended) The mobile device of claim 241, wherein based upon a determination that the digital signature is not successfully verified authenticated, the method further comprises: denying the software application ~~requesting access to the sensitive API is denied~~ access to the sensitive API.

247. (Currently Amended) 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 successfully verified authenticated, the ~~application execution manager purges the~~ method further comprises: purging the software application ~~requesting access to the sensitive API from the mobile device~~.

248. (Currently Amended) 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 access the sensitive API ~~interact with the application platform.~~

249. (Currently Amended) The method of claim 241, wherein at least some operations of the method are performed by an application execution manager, and wherein the application execution manager is implemented by a virtual machine (VM) of the mobile device.

250. (Currently Amended) The method of claim 241, wherein the digital signature is generated by applying [[a]] the private key to a first hash of the software application; and the digital signature is verified 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. (Currently Amended) The method of claim 241, wherein the digital signature is generated by applying [[a]] the private key to a first abridged version of the software application; and the digital signature is verified 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. (Currently Amended) The method of claim 241, further comprising:
determining by a code signing authority, whether the software application should be given access to the [[a]] sensitive API, wherein based upon a determination that the software application should be given access to [[a]] the sensitive API, the code signing authority accepts the software application and generates [[a]] the digital signature that is included with the software application.

253. (Currently Amended) The method of claim 241, ~~wherein the operations for controlling access to the application platform~~ further comprising~~ing~~[[e]]:
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. (Currently Amended) 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. (Currently Amended) 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. (Currently Amended) The method of claim 241, wherein the software application further includes a signature identification, and wherein the digital signature and the signature identification correspond to a mobile device type.

265. (Currently Amended) The method of claim 241, further comprising ~~wherein associating the sensitive API with the public key includes~~ obtaining the public key from a public key repository.

266. (New) The device of claim 166, wherein verifying the digital signature comprises:
hashing the software application to obtain a generated hash;
applying the public key to the digital signature to obtain a recovered hash; and
comparing the generated hash and the recovered hash.

267. (New) The system of claim 191, wherein verifying the digital signature comprises:
hashing the software application to obtain a generated hash;
applying the public key to the digital signature to obtain a recovered hash; and
comparing the generated hash and the recovered hash.

268. (New) The computer-readable storage medium of claim 216, wherein verifying the digital signature comprises:
hashing the software application to obtain a generated hash;
applying the public key to the digital signature to obtain a recovered hash; and

comparing the generated hash and the recovered hash.

269. (New) The method of claim 241, wherein verifying the digital signature comprises:
hashing the software application to obtain a generated hash;
applying the public key to the digital signature to obtain a recovered hash; and
comparing the generated hash and the recovered hash.

270. (New) The device of claim 166, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

271. (New) The system of claim 191, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

272. (New) The computer-readable storage medium of claim 216, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

273. (New) The method of claim 241, wherein the plurality of APIs comprises a plurality of sensitive APIs, wherein for each of the plurality of sensitive APIs, the mobile device allows access to the sensitive API upon verification of a digital signature unique to the sensitive API.

274. (New) The device of claim 166, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

Appl. No. 10/381,219
Amendment dated November 11, 2011
Reply to office action of May 13, 2011

275. (New) The system of claim 191, wherein the operations further comprise: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

276. (New) The computer-readable storage medium of claim 216, wherein the method further comprises: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

277. (New) The method of claim 241, further comprising: upon verifying the digital signature at the mobile device, the mobile device allowing the software application access to at least one non-sensitive API.

REMARKS/ARGUMENTS

This letter is responsive to the office action dated May 13, 2011.

Claim Amendments

In order to expedite prosecution of the application, and without prejudice, the claims have been amended to better clarify the distinguishing features of a number of Applicants' embodiments from the subject matter disclosed in the cited references. In particular, the claims have been amended to clarify that certain operations are performed at the mobile device. Other amendments have been made to enhance clarity and consistency throughout the claims.

New claims 266-269 have been added, based on features described at, for example, paragraph 28 of the published application. New claims 270-273 have been added, based on features described at, for example, paragraph 37 of the published application. New claims 274-277 have been added, based on features described at, for example, paragraph 34 and Figure 3 of the published application.

Accordingly, **claims 166-277** remain pending in this application, of which claims 166, 191, 216, and 241 are independent.

The Objection to the Specification Should Be Withdrawn

Applicants submit that the specification discloses a number of examples of computer-readable storage media (see e.g. Flash Memory 624 and RAM 626 of FIG. 6). Furthermore, the computer-readable storage medium claims recite non-transitory media, thereby excluding non-statutory media. Withdrawal of the objection is respectfully requested.

The Rejections Under 35 U.S.C. § 103 Should Be Withdrawn

The Examiner has rejected claims 166-168, 170, 171, 173-193, 195, 196, 198-218, 220, 221, 223-243, 245, 246, and 248-265 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,795,919 (hereinafter "Gibbs") in view of U.S. Patent No. 6,587,837 (hereinafter "Spagna"). The Examiner also rejected claims 169, 172, 194, 197, 219, 222, 244 and 247 under 35 U.S.C. §103(a) as being unpatentable over Gibbs and Spagna, and further in view of U.S. Patent No. 6,697,948 (hereinafter "Rabin"). The Applicants respectfully traverse all rejections.

To expedite prosecution of the application, the independent claims have been amended to clarify that, with the exception of the generation of the digital signature using a private key that is not accessible to the mobile device, a number of other acts associated with processing a request by a software application to access a sensitive API are **all performed at the mobile device**. In particular, the independent claims clarify that:

- The application platform to which access is controlled is an application platform **of the mobile device**;
- The plurality of APIs are stored **at the mobile device**;
- The indication that a software application **on the mobile device** is requesting access to a sensitive API **stored at the mobile device** is received **at the mobile device**;
- **The mobile device** uses the public key to verify the digital signature of the software application; and
- Based upon verifying the digital signature **at the mobile device, the mobile device** allows the software application access to the sensitive API.

The Examiner primarily relies on Gibbs to disclose many of the features of the independent claims, prior to the present amendment. Gibbs, however, fails to disclose the features of Applicants' independent claims as amended.

Gibbs fails to disclose restricting access to a sensitive API; Gibbs also fails to teach that the API is one of a plurality of APIs at a mobile device and the digital signature of the software application requesting access to the API was generated on a remote device; rather, Gibbs teaches that the digital signature is generated at the same device at which an API is provided

For example, at page 2 of the office action, the Examiner indicates that the claimed API is disclosed in Gibbs at column 5, lines 42-45. However, this excerpt merely indicates that an API may be provided to provide access to certain message server services. On the one hand, Gibbs teaches that the API is for the message server. On the other hand, at column 10, lines 31-62, Gibbs envisions that a digital signature is generated at the message server, and that the digital signature may then be sent to a remote user.

In this regard, the teachings of Gibbs are wholly inconsistent with the language of the amended independent claims. This is not surprising since Gibbs works in a very different way compared to the system taught by the Applicants.

First, if the claimed "sensitive API" were analogous to the Gibbs message server API, then Gibbs cannot also teach the claimed digital signature generated by a private key. It appears that Gibbs teaches that the digital signature is **also generated at the message server** – the same device that provides the API. However, Applicants' amended claims explicitly recite that although the API is at the mobile device, the private key is not accessible to the "mobile device". Even if the Examiner were of the view that Gibbs' "message server" was analogous to Applicants' "mobile device" (a position which the Applicants traverses), Gibbs would not be able to generate the digital signature without access to the private key. Therefore, since Gibbs does not teach that the API is at a mobile device while the private key used to generate the digital signature is not accessible to the mobile device, it follows that Gibbs fails to teach the corresponding elements of the amended claim.

Second, the teachings at column 10 of Gibbs suggests that the “unique digital signature” is sent to devices remote from the message server 428 (Fig. 4), such as laptop 452. Gibbs also teaches interactions with other devices, such as a WWW server 424, and a computer 404, when processing the digital signature and voting selections. It appears that Gibbs teaches a voting or polling system that involves processing that is distributed over various servers and computing devices. This is very different from Applicants’ claimed embodiments. As noted above, Applicants’ amended claims clarify that with the exception of the generation of the digital signature for a software application, most of the claimed operations are performed *on the same device*. The sensitive API to which access is being requested is also on *that same device*. Furthermore, that device is, specifically, a mobile device. It is respectfully submitted that Gibbs fails to disclose all of the features recited in the amended independent claims.

Gibbs fails to disclose a private key-public key pair; moreover, Gibbs teaches away from the use of a private key-public key pair and therefore cannot be combined with Spagna

For completeness, Applicants note that at page 5 of the office action, the Examiner concedes that Gibbs does not disclose features pertaining to a public key, but asserts that Spagna discloses a sensitive API associated with a public key. Columns 16-17 of Spagna generally teaches that a public key is employed in the use of digital signatures (and digital certificates), and columns 46-47 merely refer to “SC(s)”. **An SC is not an API**. At best, column 42 teaches that “the interface to the packer for building a SC(s) is done by APIs”. However, references made to the public key are in a context independent of the discussion of the APIs. In this regard, it is respectfully submitted that the skilled person would not be motivated to combine the teachings of Gibbs with the teachings of Spagna.

Furthermore, consider, for example, column 1 lines 59-61 of Gibbs. One of the motivations of Gibbs’ system appears to be directed towards overcoming problems with

persistent userid/password pairs which could lead to unauthorized use. However, simply incorporating the use of Spagna's private key would not address these problems, as even private keys can be shared, leaving the issue purported to be addressed by Gibbs unresolved. More generally, the teachings of Gibbs and Spagna appear to be incompatible, and it is respectfully submitted that the skilled person would not consider combining these two references, notwithstanding the fact that the Gibbs-Spagna combination fails to teach all of the features of the independent claims, as amended.

In view of at least the foregoing, the Applicants respectfully submit that the subject matter of the independent claims is neither taught nor suggested by any of the cited references, taken alone or in combination. For at least this reason, the Applicants submit that the independent claims are directed to subject matter that is both novel and non-obvious. It is further submitted that the subject matter of the dependent claims that remain in the application is also patentable for at least the same reasons. Withdrawal of the Examiner's remaining rejections under 35 U.S.C. § 103(a) is respectfully requested.

Concluding Remarks

Although the above amendments and remarks address all of the Examiner's current rejections, Applicants do not waive the right to point out that any purported impetus to combine elements from disparate references must meet established legal standards. For example, the impetus must be supported by evidence and articulated reasoning. E.g., *In re Lee*, 61 USPQ2d 1430, 1433 (Fed. Cir. 2002); see also *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (“[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness”) (quoted in *KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1741 (2007)). The evidence must suggest the desirability of the combination, not merely the feasibility. *In re Fulton*, 73 USPQ2d 1141, 1145 (Fed. Cir. 2004). Furthermore, any impetus to combine elements from disparate references must “clearly and particularly” lead one of ordinary skill in the


Appl. No. 10/381,219
Amendment dated November 11, 2011
Reply to office action of May 13, 2011

art to make a combination. See *Ruiz v. A.B. Chance Co.*, 234 F.3d 654, 660 (Fed. Cir. 2000).

In view of the foregoing remarks, Applicants respectfully submit that each of claims **166-277** is now in form for allowance. The Examiner is welcome to contact the newly-appointed Applicants' Representative to resolve any remaining issues, should the Examiner wish to expedite prosecution of the application.

Respectfully submitted,

BERESKIN & PARR LLP/S.E.N.C.R.L., s.r.l.

By  _____
Kendrick Lo
Reg. No. 54,948
Tel: 416-364-7311

Bereskin & Parr

INTELLECTUAL PROPERTY LAW

November 11, 2011

Kendrick Lo B.A.Sc. (Eng. Sci.), MBA, LL.B.
416 957 1685 klo@bereskinparr.com

Your Reference: 10/381,219
Our Reference: 13210-1465/KL

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA
22313-1450

Dear Sir:

Re: U.S. Patent Application No. 10/381,219
For: SOFTWARE CODE SIGNING SYSTEM AND METHOD
Filing Date: March 20, 2003
Applicants: David P. Yach et al.

In accordance with 37 C.F.R. 1.56 and 1.97(b)(4), the Applicant hereby submits a Supplemental Information Disclosure Statement including (1) a listing, on PTO form SB/08a, of patents and other publications of which the Applicant is aware that may be considered material to patentability, and (2) a copy of foreign and the non-patent literature documents.


The filing of this statement shall be not construed as an admission that the information cited in the attached statement is, or is considered to be, material to patentability (37 CFR 1.97(h)), nor as an admission that it constitutes prior art.

Please have the document recorded against the above-mentioned application.

Respectfully submitted,

BERESKIN & PARR LLP/S.E.N.C.R.L., s.r.l.

By _____


Kendrick Lo
Reg. No. 54,948
(416) 364-7311

Bereskin & Parr LLP
Scotia Plaza, 40 King Street West, 40th Floor, Toronto, Ontario, Canada M5H 3Y2
Tel: 416.364.7311 Fax: 416.361.1398 www.bereskinparr.com

TORONTO MISSISSAUGA WATERLOO MONTRÉAL

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		13210-1465	

U.S.PATENTS						Remove
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					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS						Remove
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. Add

FOREIGN PATENT DOCUMENTS								Remove
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² j	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	1541350	CN		2004-10-27			<input type="checkbox"/>
	2	101714201	CN		2011-05-26			<input type="checkbox"/>
	3	101694688	CN		2010-05-26			<input type="checkbox"/>

**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	13210-1465	

4	1320795	EP		2005-11-16	YACH et al.		<input type="checkbox"/>
5	2306259	EP		2011-04-06	YACH et al.		<input type="checkbox"/>
6	1626324	EP		2006-02-15	YACH et al.		<input type="checkbox"/>
7	2284644	EP		2011-02-16	YACH et al.		<input type="checkbox"/>
8	2278429	EP		2011-01-26	YACH et al.		<input type="checkbox"/>
9	2306260	EP		2011-04-06	YACH et al.		<input type="checkbox"/>
10	1626325	EP		2010-09-01	YACH et al.		<input type="checkbox"/>
11	1626326	EP		2010-09-01	YACH et al.		<input type="checkbox"/>
12	1091666	HK		2007-01-26	YACH et al.	Abstract	<input checked="" type="checkbox"/>
13	1055629	HK		2006-05-04	YACH et al.		<input type="checkbox"/>
14	1091665	HK		2010-11-19	YACH et al.		<input type="checkbox"/>

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	13210-1465		

	15	1091667	HK		2010-11-19	YACH et al.	<input type="checkbox"/>
	16	100573402	CN		2009-12-23		<input type="checkbox"/>

If you wish to add additional Foreign Patent Document citation information please click the Add button **Add**

NON-PATENT LITERATURE DOCUMENTS Remove

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	Notice of Abandonment. Canadian Application No. 2,422,917. Dated: June 20, 2011.	<input type="checkbox"/>
	2	First Office Action. Chinese Application No. 200910207911.0. Dated: August 10, 2011.	<input type="checkbox"/>
	3	Extended European Search Report. European Application No. 10186194.6. Dated: June 22, 2011.	<input type="checkbox"/>
	4	Communication Pursuant to Rules 70(2) and 70a(2) and Reference to Rule 39(1) EPC. European Application No. 10186194.6. Dated: July 25, 2011.	<input type="checkbox"/>
	5	Communication Pursuant to Article 94(3) EPC. European Application No. 10183655.9. Dated: February 23, 2011.	<input type="checkbox"/>
	6	Communication Pursuant to Article 94(3) EPC. European Application No. 10183655.9. Dated: July 13, 2011.	<input type="checkbox"/>
	7	Extended European Search Report (EESR). European Application No. 10183997.5. Dated: December 12, 2010.	<input type="checkbox"/>

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		13210-1465

8	Communication Pursuant to Article 94(3) EPC. European Application No. 10183997.5. Dated: February 23, 2011.	<input type="checkbox"/>
9	Communication Pursuant to Article 94(3) EPC. European Application No. 10183997.5. Dated: July 14, 2011.	<input type="checkbox"/>
10	Extended European Search Report. European Application No. 10186296.9. Dated: June 22, 2011.	<input type="checkbox"/>
11	Communication Pursuant to Rules 70(2) and 70a(2) and Reference to Rule 39(1) EPC. European Application No. 10186296.9. Dated: July 25, 2011.	<input type="checkbox"/>
12	Invitation pursuant to Article 94(3) and Rule 71(1) EPC dated September 28, 2011, European Patent Application No. 10186296.9.	<input type="checkbox"/>
13	First Office Action. Chinese Application No. 200910209311.8. Dated: October 19, 2011.	<input type="checkbox"/>
14	Chinese Office Action dated September 8, 2011, Chinese Patent Application No. 200910207912.5.	<input type="checkbox"/>

If you wish to add additional non-patent literature document citation information please click the Add button **Add**

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	13210-1465

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.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

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	/Kendrick Lo/	Date (YYYY-MM-DD)	2011-11-11
Name/Print	Kendrick Lo	Registration Number	54,948

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.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



[12] 发明专利申请公开说明书

[21] 申请号 01819200.9

[43] 公开日 2004 年 10 月 27 日

[11] 公开号 CN 1541350A

[22] 申请日 2001.9.20 [21] 申请号 01819200.9
 [30] 优先权
 [32] 2000. 9. 21 [33] US [31] 60/234,152
 [32] 2000. 9. 26 [33] US [31] 60/235,354
 [32] 2001. 2. 20 [33] US [31] 60/270,663
 [86] 国际申请 PCT/CA2001/001344 2001.9.20
 [87] 国际公布 WO2002/025409 英 2002.3.28
 [85] 进入国家阶段日期 2003.5.20
 [71] 申请人 捷讯研究有限公司
 地址 加拿大安大略省
 [72] 发明人 戴维·P·亚切
 迈克尔斯·S·布朗
 赫伯特·A·利特尔

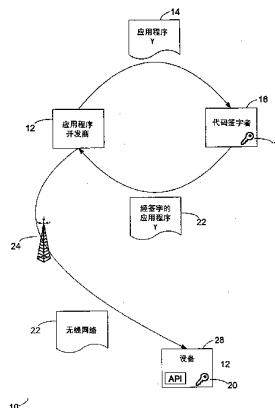
[74] 专利代理机构 中科专利商标代理有限责任公
 司
 代理人 戎志敏

权利要求书 7 页 说明书 16 页 附图 7 页

[54] 发明名称 代码签字系统和方法

[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。



ISSN 1008-4274

1. 一种代码签字系统，用于与具有数字签字的软件应用程序一起工作，包括：

应用平台；

应用编程接口（API），用来把软件应用程序和应用平台链接；

虚拟机，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。

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，虚拟机否定软件应用程序访问 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，虚拟机消除该软件应用程序。

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. 一种代码签字系统，用于与具有数字签字的软件应用程序一起工作，包括：

应用平台；

一组应用编程接口（API），每个 API 用来把软件应用程序与应用平台上的资源连接；

虚拟机，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，为控制软件应用程序对多个 API 的访问，虚拟机验证数字签字的真实性。

7. 根据权利要求 6 所述的代码签字系统，其特征在于多个 API 包括在 API 程序库内。

8. 根据权利要求 6 所述的代码签字系统，其特征在于一个或多个 API 被分类为敏感的 API，其中虚拟机用数字签字来控制访问敏感的 API。

9. 根据权利要求 8 所述的代码签字系统，其特征在于：用于与多个软件应用程序一起工作，其中一个或多个软件应用程序具有数字签字，

而且，为了控制多个软件应用程序中的每个访问敏感的 API，其中的虚拟机要验证数字签字的真实性。

10. 根据权利要求 6 所述的代码签字系统，其特征在于应用平台上的资源包括无线通信系统。

11. 根据权利要求 6 所述的代码签字系统，其特征在于应用平台上的资源包括实现加密算法的加密模块。

12. 根据权利要求 6 所述的代码签字系统，其特征在于应用平台上的资源包括数据存储器。

13. 根据权利要求 6 所述的代码签字系统，其特征在于应用平台上的资源包括用户接口 (UI)。

14. 根据权利要求 1 所述的代码签字系统，其特征在于还包括：

一组 API 程序库，每个包括一组 API，其中虚拟机控制软件应用程序对多个 API 程序库的访问。

15. 根据权利要求 14 所述的代码签字系统，其特征在于一组或多于一组的 API 程序库分类成敏感的 API 程序库，并且其中的虚拟机用数字签字来控制软件应用程序访问敏感的 API 程序库组。

16. 根据权利要求 15 所述的代码签字系统，其特征在于软件应用程序包括对每个敏感的 API 程序库的唯一数字签字。

17. 根据权利要求 16 所述的代码签字系统，其特征在于软件应用程序包括对每个唯一的数字签字的签字标识；

每个敏感的 API 程序库包括签字标识符；

虚拟机比较签字标识与签字标识符，来使唯一的数字签字与敏感的 API 程序库匹配。

18. 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用专用签字密钥产生，且虚拟机用公用签字密钥来验证数字签字的真实性。

19. 根据权利要求 18 所述的代码签字系统，其特征在于：

数字签字借助于把专用签字密钥作用于软件应用程序的杂乱信号来产生；

虚拟机用产生软件应用程序杂乱信号来获得被产生的杂乱信号 (hash)，并把公用签字密钥作用于数字签字以获得恢复的杂乱信号并把

产生的杂乱信号与恢复的杂乱信号相比较来验证数字签名的真实性。

20. 根据权利要求 1 所述的代码签字系统，其特征在于 API 进一步包括：

描述字符串，当软件应用程序要想访问 API 时，它由移动设备显示。

21. 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。

22. 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一个或多个移动设备的核心功能。

23. 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移动设备上的硬件。

24. 根据权利要求 23 所述的代码签字系统，其特征在于硬件包括用户身份模块（SIM）卡。

25. 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 应用软件。

26. 根据权利要求 1 所述的代码签字系统，其特征在于 API 接口于应用平台上的加密程序。

27. 根据权利要求 1 所述的代码签字系统，其特征在于 API 与应用平台上的专有数据模型接口。

28. 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。

29. 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要访问敏感的应用程序编程接口（API）的移动设备上；

确定软件应用程序是否包括与敏感的 API 有关的数据签字；

如要软件应用程序不包括与敏感的 API 有关的数字签字，那么否定软件应用程序访问敏感的 API。

30. 根据权利要求 29 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与敏感的 API 有关的数字签字，那么从移动设备上消除软件应用程序。

31. 根据权利要求 29 所述的方法，其特征在于数字签字由代码签字授权机构产生。

32. 根据权利要求 29 所述的方法，其特征在于包括附加的步骤：

如果软件应用程序包括与敏感的 API 有关的数字签字，那么验证数字签字的真实性；

如果数字签字不真实，则否定软件应用程序访问敏感的 API。

33. 根据权利要求 32 所述的方法，其特征在于包括附加的步骤：

如果数字签字不真实，则从移动设备上清除该软件应用程序。

34. 根据权利要求 32 所述的方法，其特征在于数字签字是借助于把专用签字密钥应用于软件应用程序的杂乱信号来产生，并且其中验证数字签字真实性的方法由包括下列步骤的方法来执行：

把相应于专用签字密钥的公用签字密钥存到移动设备上；

产生软件应用程序的杂乱信号以获得被产生的杂乱信号；

把公用签字密钥作用于数字签字密钥，以获得恢复的杂乱信号；和

把被产生的杂乱信号与恢复的杂乱信号相比较。

35. 根据权利要求 34 所述的方法，其特征在于数字签字由计算软件应用程序的杂乱信号和应用专用签字密钥来产生。

36. 根据权利要求 29 所述的方法，其特征在于包括如下步骤：

显示描述字符串，它通知移动设备的用户，软件应用程序要访问敏感的 API。

37. 根据权利要求 36 所述的方法，其特征在于包括附加的步骤：

从用户接收指令，准许或否定软件应用程序访问敏感的 API。

38. 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；

评审软件应用程序，确定它是否可访问 API；

如果软件应用程序可以访问 API，则把数字签字添加到软件应用程序；

验证添加到软件应用程序的数字签字的真实性；

对添加的数字签字是真实的软件应用程序提供访问 API。

39. 根据权利要求 38 所述的方法，其特征在于评审软件应用程序的步骤是由代码签字授权机构执行的。

40. 根据权利要求 38 所述的方法，其特征在于把数字签字添加到软件应用程序的方法是由包括下列步骤的方法执行的：

计算软件应用程序的杂乱信号；

把签字密钥应用于软件应用程序的杂乱信号，以产生数字签字。

41. 根据权利要求 40 所述的方法，其特征在于软件应用程序的杂乱信号 (hash) 用安全的杂乱信号算法 (SHA1) 计算。

42. 根据权利要求 40 所述的方法，其特征在于验证数字签字真实性的方法包括：

在移动设备上提供相应的签字密钥；

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；

把相应的签字密钥应用于数字签字，以获得恢复的杂乱信号；

借助于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。

43. 根据权利要求 42 所述的方法，其特征在于进一步包括，如果数字签字不真实，则否定该软件应用程序访问 API。

44. 根据权利要求 42 所述的方法，其特征在于签字密钥是专用签字密钥，其相应的签字密钥是公用签字密钥。

45. 一种在移动设备上控制访问敏感 API 的方法，包括：

注册一个或多个软件开发商，其设计的访问敏感的 API 的软件应用程序是可信的；

接收软件应用程序的杂乱信号；

确定软件应用程序是否是注册的软件开发商所设计；

如果是注册的软件开发商所设计，则用软件应用程序的杂乱信号产生数字签字，其中，

数字签字可添加到软件应用程序；

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。

46. 根据权利要求 45 所述的方法，其特征在于产生数字签字的步骤是由代码签字授权机构执行的。

47. 根据权利要求 45 所述的方法，其特征在于产生数字签字的步骤是把签字密钥应用于软件应用程序的杂乱信号产生的。

48. 根据权利要求 47 所述的方法，其特征在于移动设备执行下列附加的步骤验证数字签字的真实性；

在移动设备上提供相应的签字密钥；

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；

把相应的签字密钥应用于数字签字，以获得恢复的杂乱信号；

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，则否定软件应用程序访问敏感的 API。

49. 一种在移动设备上限制访问 API 的方法，包括如下步骤：

把软件应用程序装到要访问一个或多个 API 的移动设备上；

确定软件应用程序是否包括与移动设备相关的真实数字签字；

如果软件应用程序不包括与移动设备有关的真实数字签字，则否定软件应用程序访问一个或多个 API。

50. 根据权利要求 49 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与移动设备有关的真实数字签字，则从移动设备上消除该软件应用程序。

51. 根据权利要求 49 所述的方法，其特征在于：

软件应用程序包括一组数字签字；

该组数字签字包括与各不同的移动设备类型有关的数字签字。

52. 根据权利要求 51 所述的方法，其特征在于每组数字签字是由各相应的代码签字授权机构产生的。

53. 根据权利要求 49 所述的方法，其特征在于确定软件应用程序是否包括与移动设备有关的真实数字签字的方法包括如下附加步骤：

确定软件应用程序是否包括与移动设备有关的数字签字；

如果有，则验证其真实性。

54. 根据权利要求 53 所述的方法，其特征在于一个或多个 API 包括

一个或多个分类为敏感的 API，对于每个敏感的 API，该方法进一步包括：

确定软件应用程序是否包括与敏感的 API 有关的真实数字签字；

如果软件应用程序不包括与敏感的 API 有关的真实数字签字，则否定软件应用程序访问敏感的 API。

55. 根据权利要求 52 所述的方法，其特征在于每个数字签字是由其相应的代码签字授权机构应用各专用的签字密钥于软件应用程序的杂乱信号而产生的。

56. 根据权利要求 55 所述的方法，其特征在于确定软件应用程序是否包括与移动设备有关的真实数字签字的方法包括如下步骤：

确定软件应用程序是否包括与移动设备有关的数字签字；

如果有，则验证其真实性，

其中验证数字签字真实性的方法由包括下列步骤的方法执行：

把公用密钥存在移动设备上，它相应于与代码签字授权机构有关的专用签字密钥，该签字机构产生与移动设备有关的签字；

产生软件应用程序的杂乱信号，以获得产生的杂乱信号；

把公用密钥应用于数字签字，以获得恢复的杂乱信号；和

比较产生的杂乱信号和恢复的杂乱信号。

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日申请的美国条约申请，申请号是60/234152；“代码签字系统及方法”于2000年9月22日申请的美国条约申请，申请号是60/235354；“代码签字系统及方法”于2001年2月20日申请的美国条约申请，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™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者的软件应用程序。一旦该软件被用户下载，用户必须只根据对软件开发者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。

发明内容

提供了代码签字系统和方法。代码签字系统与具有数字签字的软件一起工作，并包括应用平台，应用程序编程界面（API），和虚拟机。API 用来把软件应用程序与应用平台连接，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。

根据本发明的另一实施例，与有数字签字的软件一起工作的代码字符系统，包括应用平台，一组 API，每个 API 用来把软件应用程序与应用平台上的资源相连接，和虚拟机，它验证数字签字的真实性，以控制软件应用程序对 API 的访问，其中虚拟机验证数字签字的真实性，以控制软件应用程序对一组 API 的访问。

根据本发明的另一实施例，控制访问移动设备上敏感的应用程序编程界面的方法包括把软件应用程序装到移动设备上，该移动设备是要访问敏感的 API 的设备，确定软件应用程序是否包括与敏感的 API 有关的数字签字，如果不包括则否定该软件应用程序访问敏感的 API。

在本发明的另一实施例中，控制由软件开发商开发的软件应用程序访问在移动设备上应用程序编程界面（API）的方法包括从软件开发商接收软件应用程序，检查该软件应用程序，以确定它是否可访问 API，如果可以，则添加数字签字于软件应用程序中，验证加到软件应用程序中的数字签字的真实性，并给添加的数字签字是真实的软件应用程序提供访问 API。

根据本发明的另一实施例，限制访问移动设备上敏感的 API 的方法包括登记一个或多个可信任的开发商来设计访问敏感的 API 的软件应用程序，接收软件应用程序的杂乱信号（hash），确定软件应用程序是否由注册的软件开发商设计，如果是，则用软件应用程序的杂乱信号（hash）产生数字签字，并添加到软件应用程序中，移动设备验证数字签字的真实性，以控制该软件应用程序访问敏感的 API。

在更进一步的实施例中，限制访问移动设备上应用程序编程界面的方法包括把软件应用程序装到需访问一个或多个 API 的移动设备上，确定软件应用程序是否包括与该移动设备有关的数字签字，如果不包括，

则否定该软件应用程序访问一个或多个 API。

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；

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存储系统通信，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（如地址簿或日历本）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“全球”签字得到验证。例如，公司可能希望防止它的雇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，至少需要全球签字，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，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,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 相同的散到算法, 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。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。签字识别参考图 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，或分配到公司雇员。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，并能使虚拟机让数字签字 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 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 与签字识别符 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，设备 62E 可要求既有全球签字，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 可要求只有全球签字的验证，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球签字的一组设备可包括目标设备。既要求全球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，本地振荡器（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™ 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 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“全球”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 Bluetooth™（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。

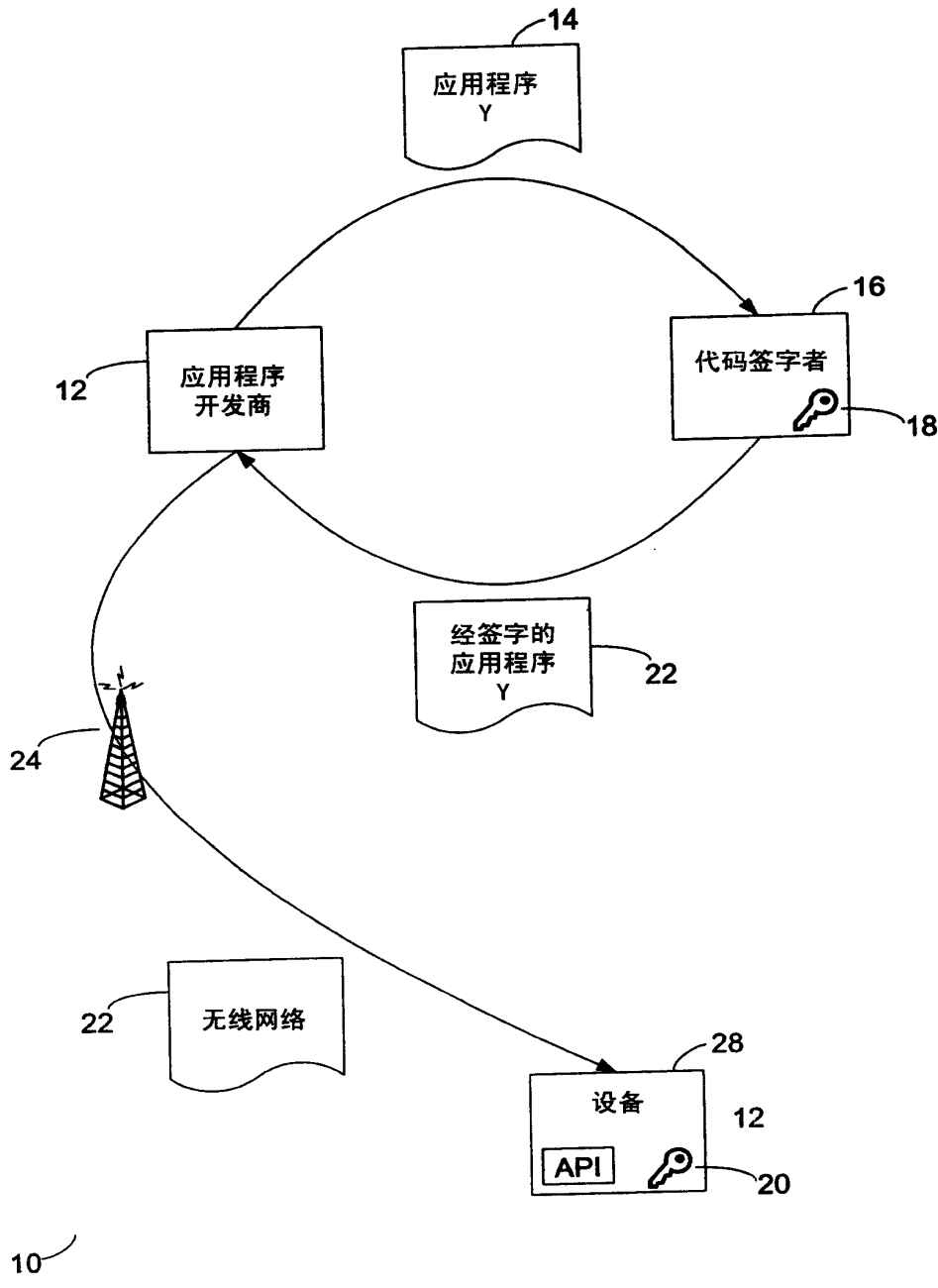


图 1

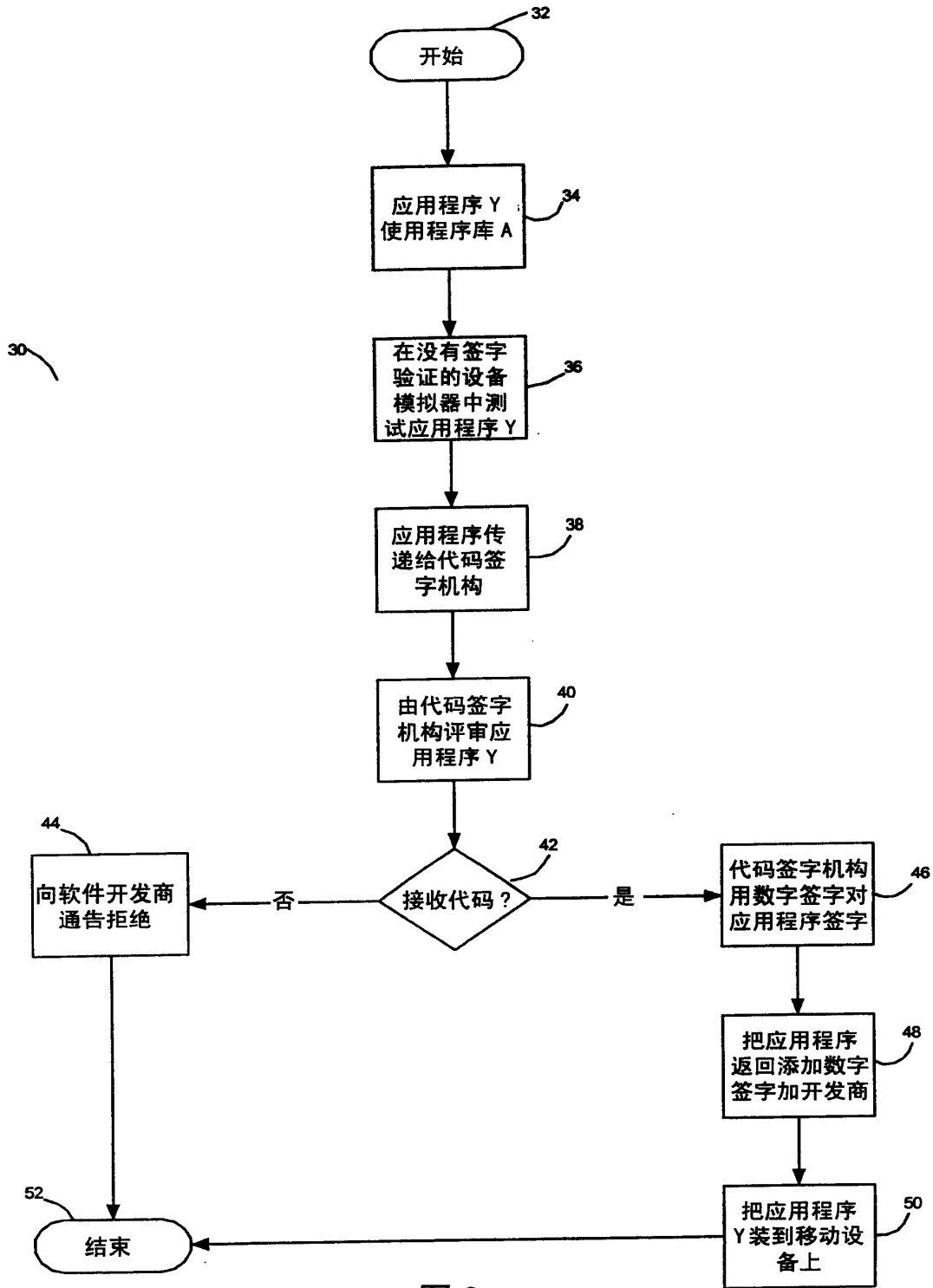
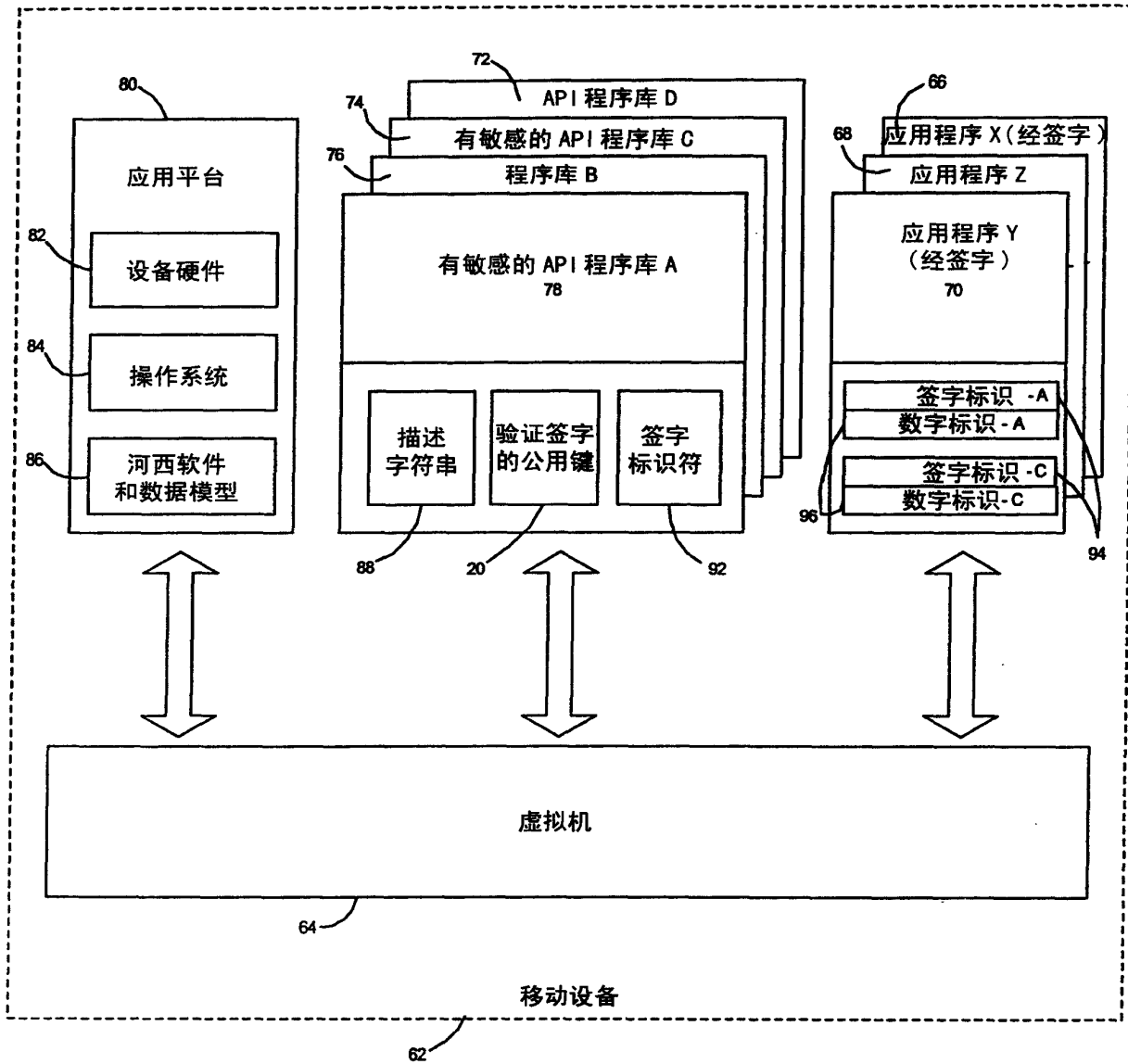


图 2



60

图 3

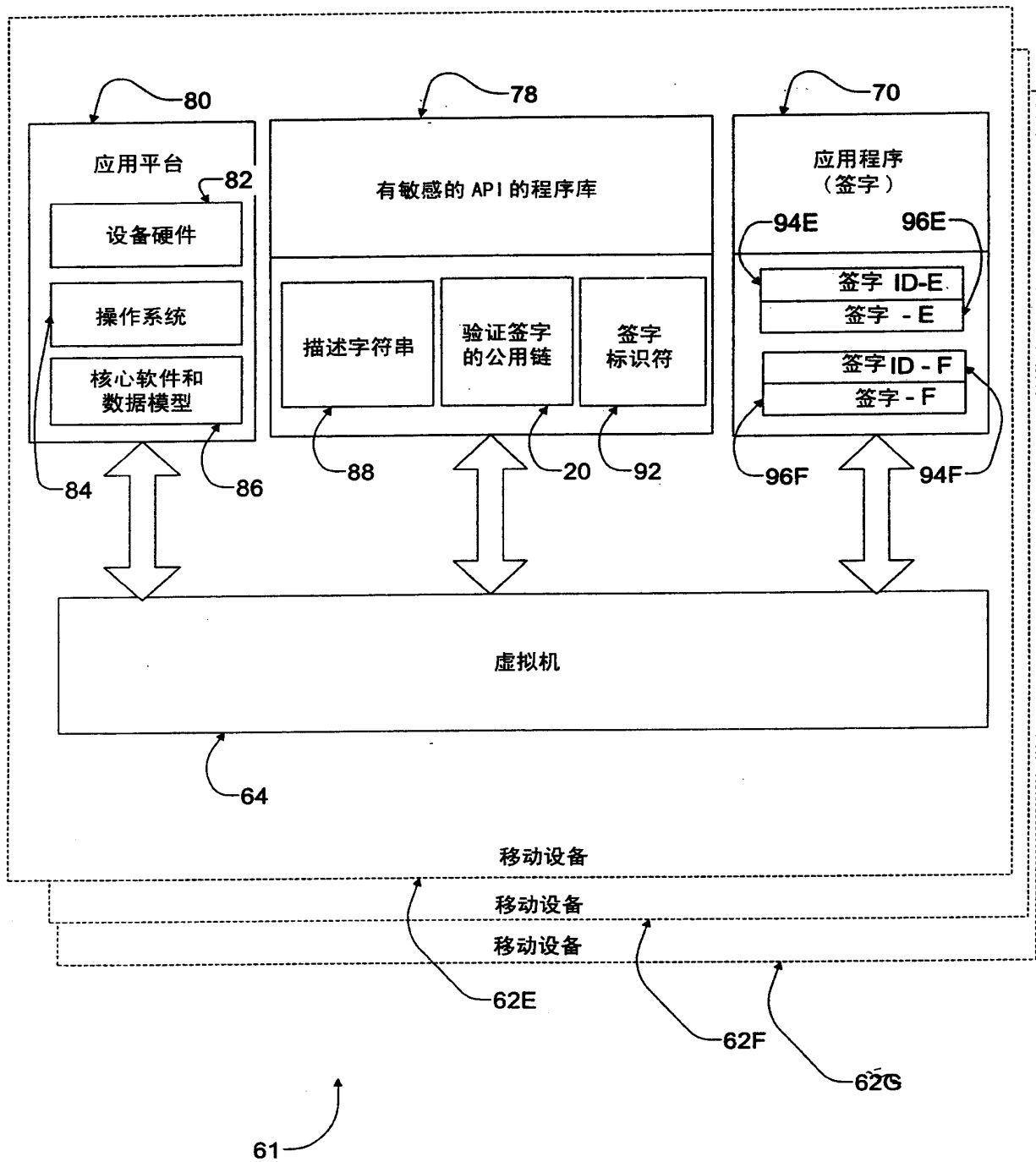
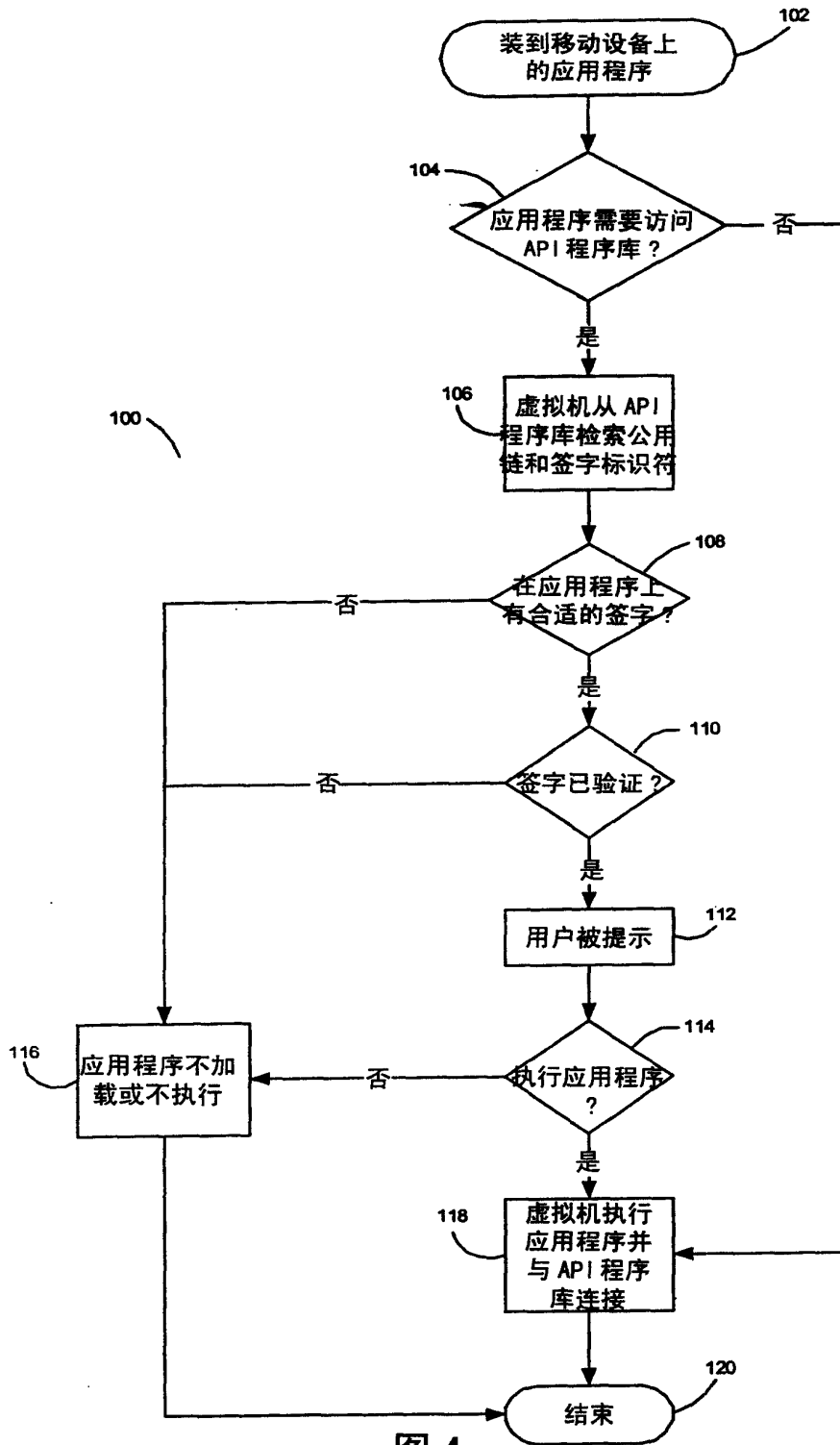


图 3A



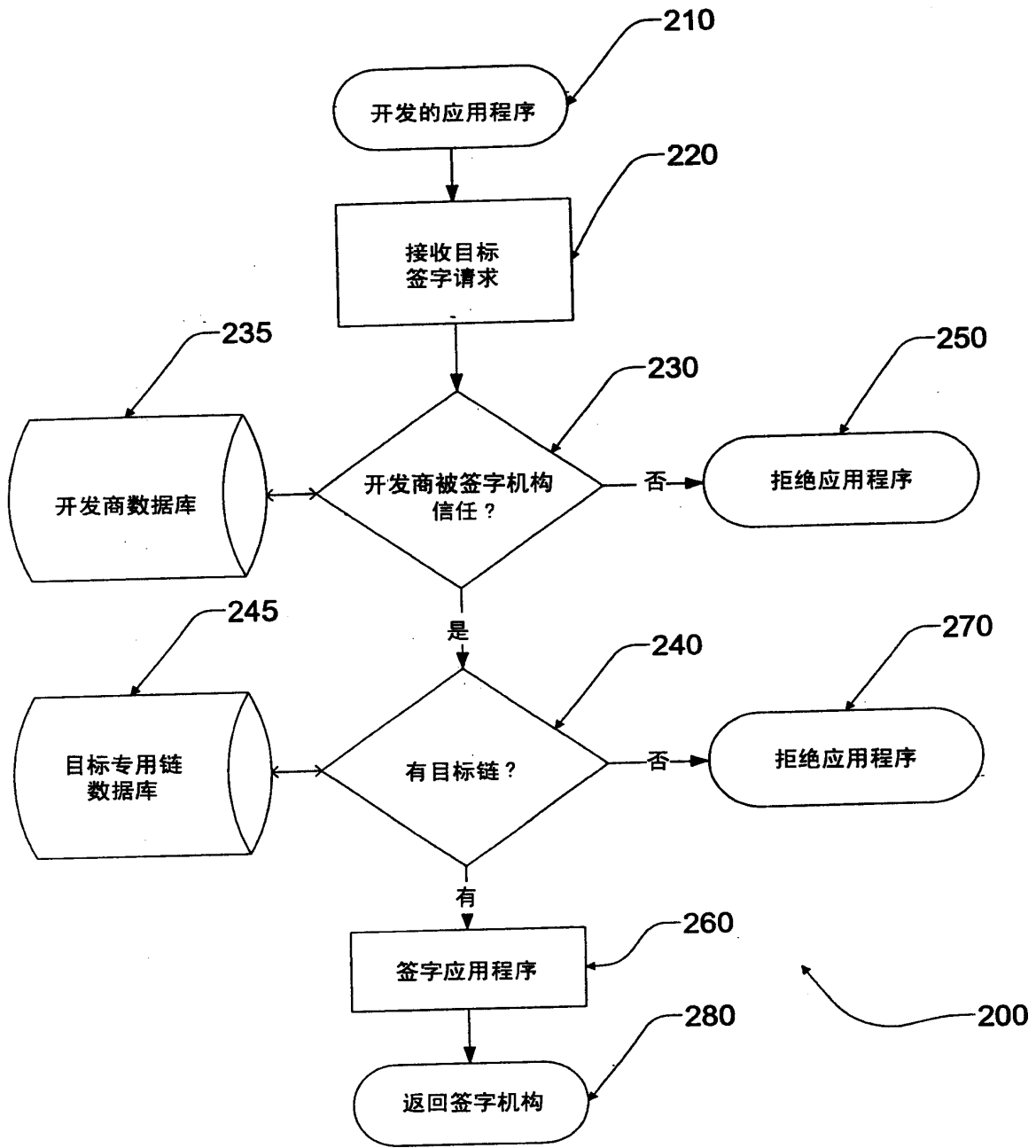
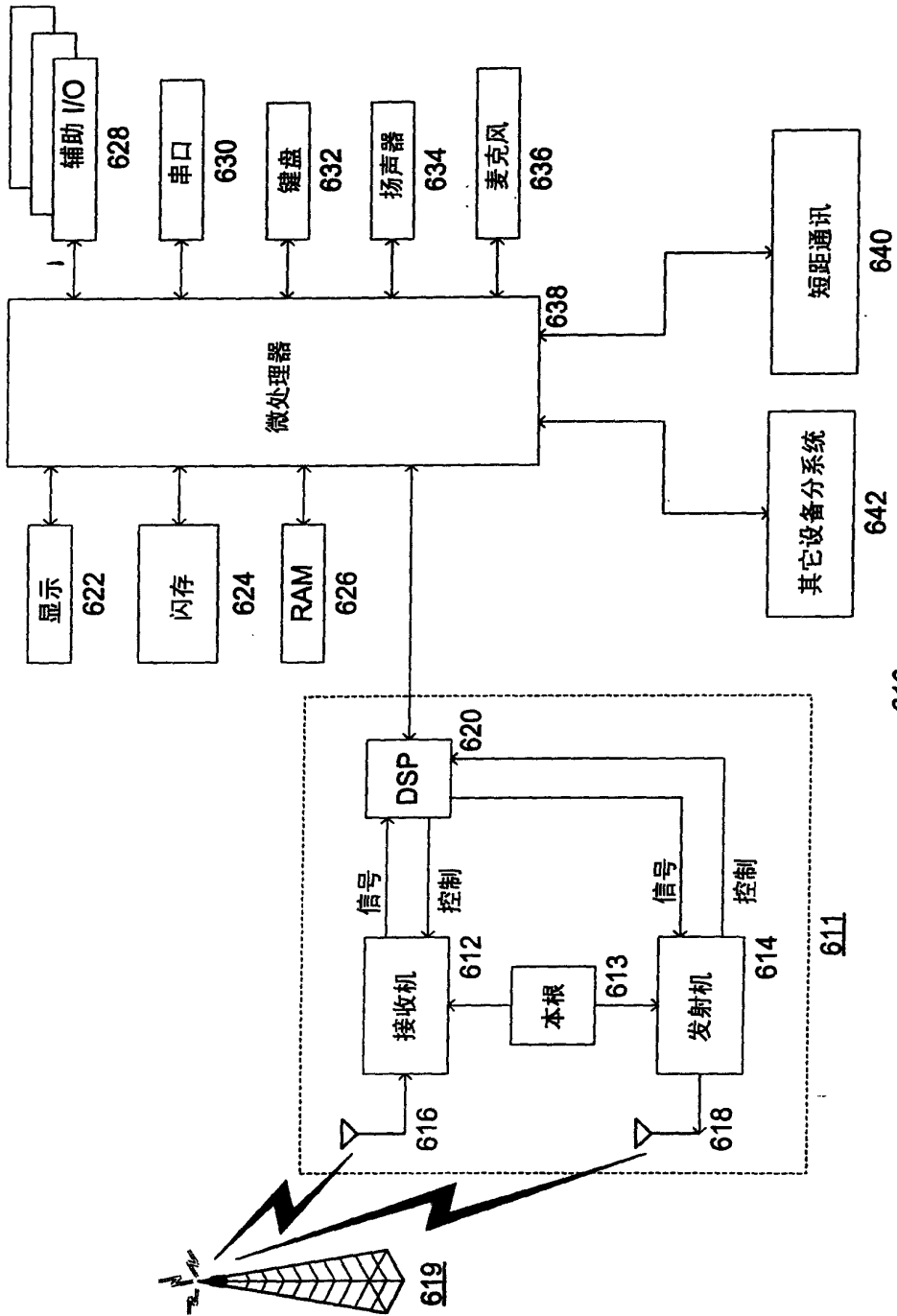


图 5



610

图 6



(12) 发明专利申请

(10) 申请公布号 CN 101714201 A

(43) 申请公布日 2010.05.26

(21) 申请号 200910209311.8

(22) 申请日 2001.09.20

(30) 优先权数据

60/234,152 2000.09.21 US

60/235,354 2000.09.26 US

60/270,663 2001.02.20 US

(62) 分案原申请数据

01819200.9 2001.09.20

(71) 申请人 捷讯研究有限公司

地址 加拿大安大略省沃特卢市

(72) 发明人 戴维·P·亚切 迈克尔斯·S·布朗

赫伯特·A·利特尔

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 戎志敏

(51) Int. Cl.

G06F 21/22 (2006.01)

H04L 29/06 (2006.01)

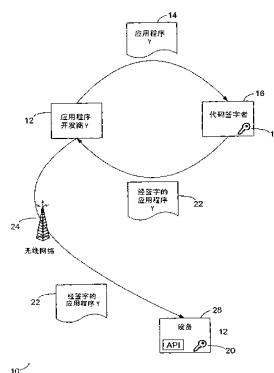
权利要求书 7 页 说明书 14 页 附图 7 页

(54) 发明名称

代码签字系统和方法

(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。



CN 101714201 A

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;

其中,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 4 所述的代码签字系统,其特征在于 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 API 时,移动设备显示描述字符串。

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,包括步骤:
把软件应用程序装载到移动设备上,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。
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至少一个 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,用于认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认证后,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认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、数据存储器和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 还包括描述字符串,其中,当软件应用程序试图访问至少一个 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;

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认证数字签字。

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,包括步骤:

把具有数字签字和签字标识的软件应用程序装载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,则认证数字签字;

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,则认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,代码签字授权机构根据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,其特征在于还包括步骤:

如果软件应用程序不包括真实的数字签字,则从移动设备上删除该软件应用程序。

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 37 所述的方法,其特征在于认证数字签字的步骤包括:

如果签字标识对应至少一个签字标识符,则验证数字签字的真实性。

43. 根据权利要求 37 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。

44. 根据权利要求 37 所述的方法,其特征在于:

移动设备包括多个 API;

至少一个 API 被分类为敏感的;

访问任一个 API 要求真实的全局签字;

访问每个敏感的 API 要求真实的全局签字和与签字标识关联的真实的数字签字;

认证数字签字包括;

确定软件应用程序要求访问的一个或多个 API 是否敏感的 API;

确定软件应用程序是否包括真实的全局签字;

确定软件应用程序是否包括真实的数字签字和签字标识,其中,软件应用程序要求访问的一个或多个 API 包括敏感的 API 和软件应用程序包括真实的全局签字;

拒绝软件应用程序访问一个或多个 API 包括:

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

如果软件应用程序要求访问的一个或多个 API 包括敏感的 API、软件应用程序包括真实的全局签字、软件应用程序不包括要求访问敏感的 API 的真实的数字签字和签字标识符,则拒绝软件应用程序访问敏感的 API。

45. 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:

如果签字标识符对应于各个 API 的签字标识符,则认证各个软件应用程序提供的数字签字来访问 API,其中,用于软件应用程序的数字签字由对应签字标识符的签字标识产生,以便访问至少一个 API;

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与签字标识关联的签字方案下的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相同来认证数字签字。

46. 根据权利要求 45 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认证,则允许访问 API 的程序库。

47. 根据权利要求 45 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。

48. 根据权利要求 45 所述的方法,其特征在于至少一个 API 还包括描述字符串,其中,当软件应用程序试图访问至少一个 API 时,向用户显示描述字符串。

49. 根据权利要求 45 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。

50. 根据权利要求 45 所述的方法,其特征在于要求软件应用程序提供全局数字签字的验证,以访问任何 API。

51. 一种在移动设备上控制访问具有签字标识符的敏感的应用程序编程接口 API 的方法,包括:

注册一个或多个可信的软件开发商,开发的软件访问敏感的 API;

接受软件应用程序的杂乱信号;

确定杂乱信号是否由注册的软件开发商发送;

如果杂乱信号是由注册的软件开发商所发送,则使用软件应用程序的杂乱信号和对应签字标识符的签字标识产生数字签字;

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,则移动设备验证数字签字的真实性,以便控制软件应用程序访问敏感的 API。

52. 根据权利要求 51 所述的方法,其特征在于由代码签字授权机构产生数字签字。

53. 根据权利要求 51 所述的方法,其特征在于通过将签字密钥应用到软件应用程序的杂乱信号产生数字签字。

54. 根据权利要求 53 所述的方法,其特征在于移动设备验证数字签字的真实性包括:

在移动设备上提供对应的签字密钥;

计算移动设备上的软件应用程序的杂乱信号,获得计算的杂乱信号;

将对应的签字密钥应用到数字签字,获得恢复的杂乱信号;

通过比较计算的杂乱信号和恢复的杂乱信号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。

55. 根据权利要求 51 所述的方法,其特征在于提供了验证数字签字的真实性的虚拟机。

56. 根据权利要求 55 所述的方法,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。

57. 根据权利要求 55 所述的方法,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。

58. 根据权利要求 55 所述的方法,其特征在于使用专用签字密钥产生数字签字,虚拟机使用公用签字密钥验证数字签字的真实性。

59. 根据权利要求 58 所述的方法,其特征在于:
通过将专用签字密钥应用到软件应用程的杂乱信号产生数字签字;
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。

代码签字系统及方法

[0001] 有关申请的参照

[0002] 本申请要求下列申请的优先权：

[0003] “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

[0004] 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™ 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

[0005] 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者的软件应用程序。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。

[0006] 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。

[0007] 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。

发明内容

[0008] 本发明的目的是提供代码签字系统和方法。

[0009] 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：

[0010] 应用平台；

[0011] 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；

[0012] 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，

[0013] 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。

- [0014] 优选地,如果数字签字不真实,则虚拟机拒绝软件应用程序访问 API。
- [0015] 优选地,如果数字签字不真实,则虚拟机删除软件应用程序。
- [0016] 优选地,代码签字系统装在移动设备上。
- [0017] 优选地,数字签字由代码签字授权机构产生。
- [0018] 优选地,还包括:
- [0019] 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。
- [0020] 优选地,至少一个 API 程序库被分类为敏感的;
- [0021] 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;
- [0022] 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,用于访问敏感的 API 程序库;
- [0023] 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。
- [0024] 优选地,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。
- [0025] 优选地,应用平台包括操作系统。
- [0026] 优选地,包括一个或多个移动设备的核心功能。
- [0027] 优选地,包括移动设备上的硬件。
- [0028] 优选地,硬件包括用户身份模块卡。
- [0029] 优选地,软件应用程序是用于移动设备的 Java 应用程序。
- [0030] 优选地,API 与应用平台上的加密流程接口。
- [0031] 优选地,API 与应用平台上的专用数据模块接口。
- [0032] 优选地,虚拟机是安装在移动设备上的 Java 虚拟机。
- [0033] 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:
 - [0034] 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;
 - [0035] 确定软件应用程序是否包括数字签字和签字标识;
 - [0036] 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;
 - [0037] 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,
 - [0038] 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:
 - [0039] 在移动设备上存储对应专用签字密钥的公用签字密钥;
 - [0040] 产生软件应用程序的杂乱信号来获得产生的杂乱信号;
 - [0041] 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;

- [0042] 比较产生的杂乱信号和恢复的杂乱信号。
- [0043] 优选地,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。
- [0044] 优选地,数字签字和签字标识由代码签字授权机构产生。
- [0045] 优选地,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。
- [0046] 优选地,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。
- [0047] 优选地,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。
- [0048] 优选地,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。
- [0049] 优选地,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。
- [0050] 按照本发明的另一方面,一种移动设备,包括:
- [0051] 应用平台,具有应用编程接口 API;
- [0052] 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;
- [0053] 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;
- [0054] 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,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设备;
- [0055] 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;
- [0056] 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;
- [0057] 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;
- [0058] 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,
- [0059] 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。
- [0060] 优选地,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。
- [0061] 优选地,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。
- [0062] 优选地,应用平台的 API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0063] 优选地,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。

- [0064] 优选地,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。
- [0065] 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:
- [0066] 从软件开发商接收软件应用程序;
- [0067] 确定软件应用程序是否满足至少一个标准;
- [0068] 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;
- [0069] 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;
- [0070] 如果数字签字是真实的,向软件应用程序提供到 API 的路径;
- [0071] 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:
- [0072] 计算软件应用程序的杂乱信号;
- [0073] 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;
- [0074] 在移动设备上提供公用签字密钥;
- [0075] 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;
- [0076] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0077] 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。
- [0078] 优选地,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。
- [0079] 优选地,使用安全的杂乱信号算法计算软件应用程序的杂乱信号。
- [0080] 优选地,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。
- [0081] 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程序编程接口 API 的方法,包括步骤:
- [0082] 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;
- [0083] 接收软件应用程序的杂乱信号;
- [0084] 确定杂乱信号是否是注册的软件开发商所发送;
- [0085] 产生数字签字,其中,
- [0086] 数字签字和签字标识被添加到软件应用程序;
- [0087] 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;
- [0088] 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;
- [0089] 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:
- [0090] 在移动设备上提供公用签字密钥;
- [0091] 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;
- [0092] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0093] 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;

- [0094] 如果数字签字不真实,则拒绝软件应用程序访问敏感的 API。
- [0095] 优选地,产生数字签字的步骤由代码签字授权机构执行。
- [0096] 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:
- [0097] 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;
- [0098] 如果签字标识对应签字标识符,则验证数字签字;
- [0099] 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;
- [0100] 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:
- [0101] 验证与签字标识符对应的签字标识;
- [0102] 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;
- [0103] 产生软件应用程序的杂乱信号,以获得产生的杂乱信号;
- [0104] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0105] 将产生的杂乱信号与恢复的杂乱信号进行比较。
- [0106] 优选地,数字签字和签字标识与移动设备的类型有关。
- [0107] 优选地,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。
- [0108] 优选地,软件应用程序包括多个数字签字和签字标识;
- [0109] 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。
- [0110] 优选地,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。
- [0111] 优选地,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。
- [0112] 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,软件应用程序具有数字签字和签字标识,包括:
- [0113] 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;
- [0114] 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;
- [0115] 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;
- [0116] 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;
- [0117] 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。
- [0118] 优选地,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。
- [0119] 优选地,API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0120] 优选地,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。

[0121] 优选地,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。

[0122] 优选地,要求软件应用程序提供全局数字签字的验证,以访问任何 API。

附图说明

[0123] 图 1 是根据本发明实施例的代码签字协议图;

[0124] 图 2 是图 1 的代码签字协议的流程图;

[0125] 图 3 是在移动设备上的代码签字系统方框图;

[0126] 图 3A 是在一组移动设备上的代码签字系统方框图;

[0127] 图 4 是图 3 和图 3A 代码签字系统的工作流程图;

[0128] 图 5 是管理图 3A 的代码签字真实性的流程图;

[0129] 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

[0130] 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 Java 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。

[0131] 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(如地址簿或日历本)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。

[0132] 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职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。

[0133] 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。

[0134] 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)。

[0135] 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,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 SHA1)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。

[0136] 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。

[0137] 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。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。

[0138] 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。

[0139] 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。

[0140] 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附加到软件应用程序 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。

[0141] 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,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。

[0142] 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,或分配到公司雇员。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。

[0143] 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。

[0144] 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。

[0145] 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,并能使虚拟机让数字签字 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。

[0146] 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 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。

[0147] 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 不于图中。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,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。

[0148] 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。

[0149] 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。

[0150] 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)。

[0151] 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。

[0152] 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。

[0153] 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。

[0154] 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。

[0155] 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。

[0156] 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)。

[0157] 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,本地振荡器(L0)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™ 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。

[0158] 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。

[0159] 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。

[0160] 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。

[0161] 设备 610 优先包括微处理器 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。

[0162] 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 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。

[0163] 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。

[0164] 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。

[0165] 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。

[0166] 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。

[0167] 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。

[0168] 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 Bluetooth™(蓝牙)通信模式,以提供与有相同能力的系统和设备通信。

[0169] 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。

[0170] 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。

[0171] 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。

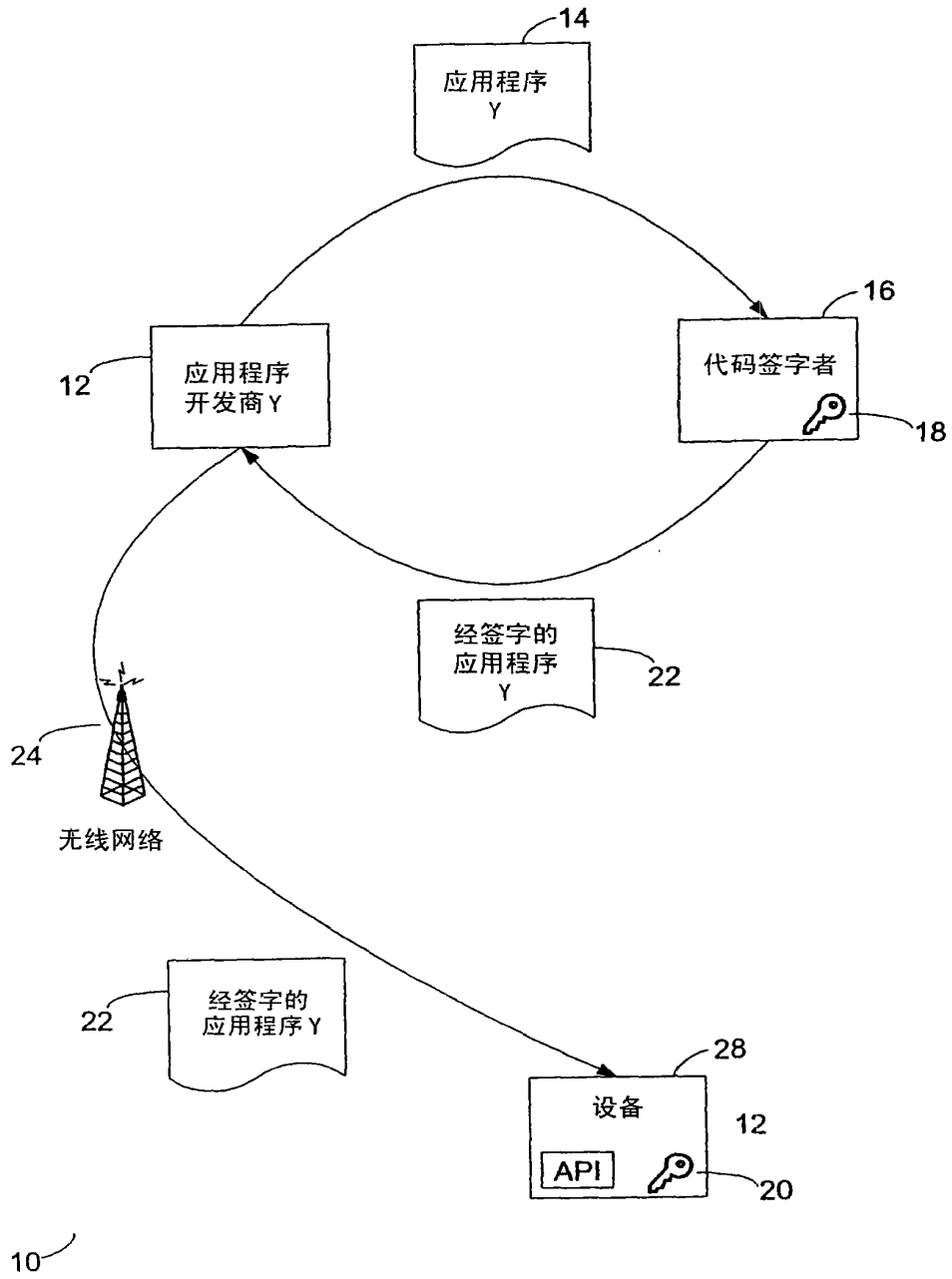


图 1

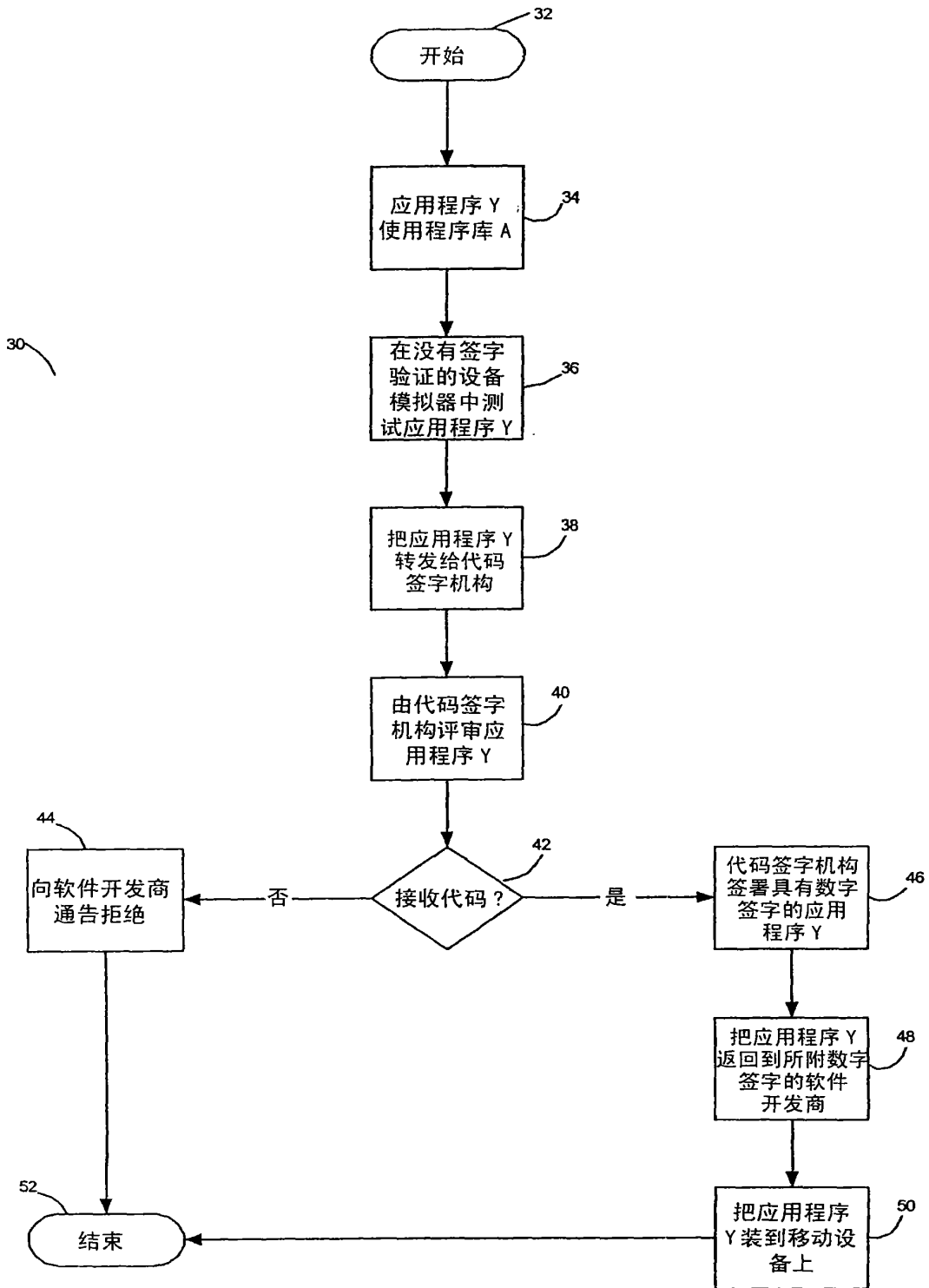
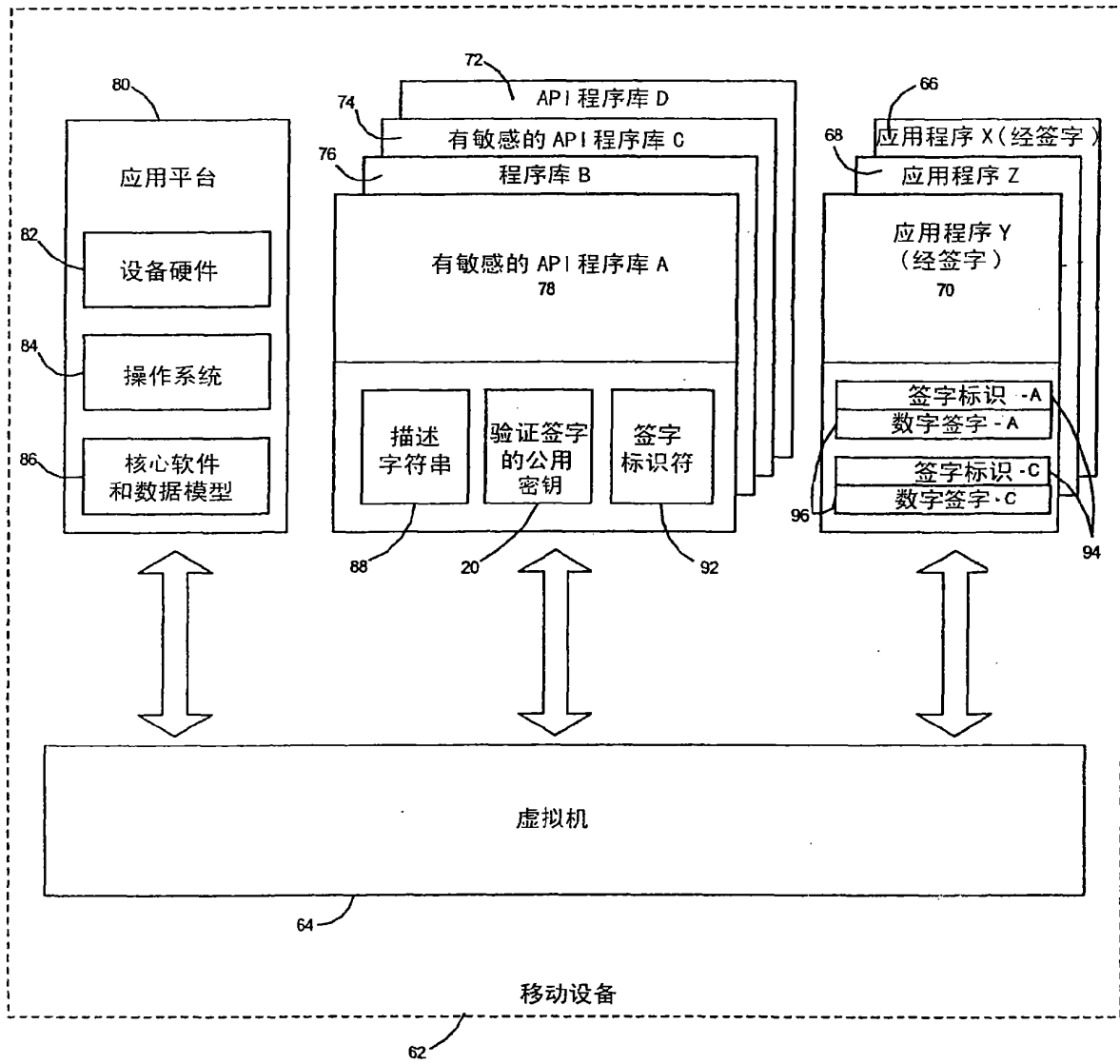


图 2



60

图 3

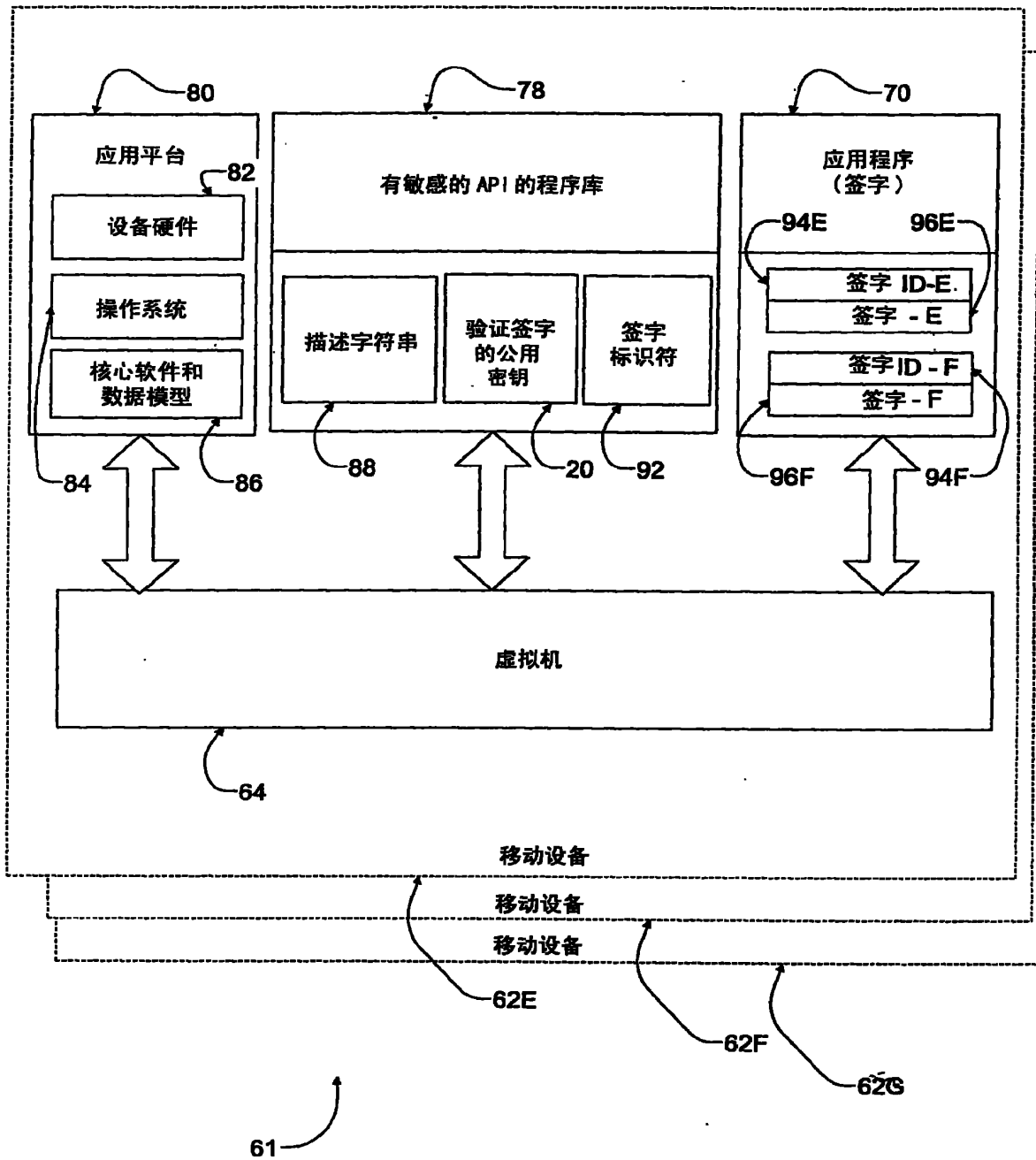


图 3A

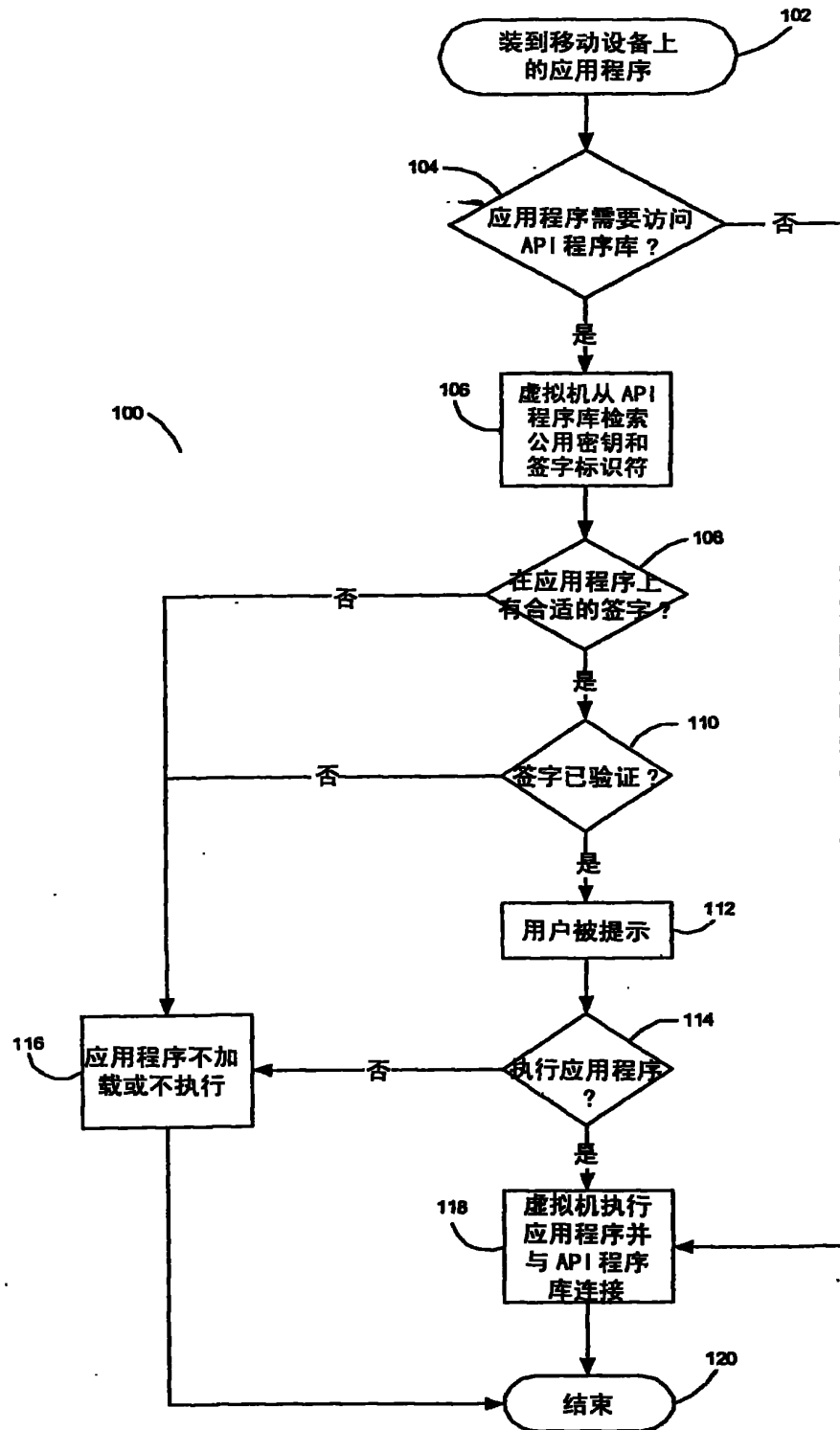


图 4

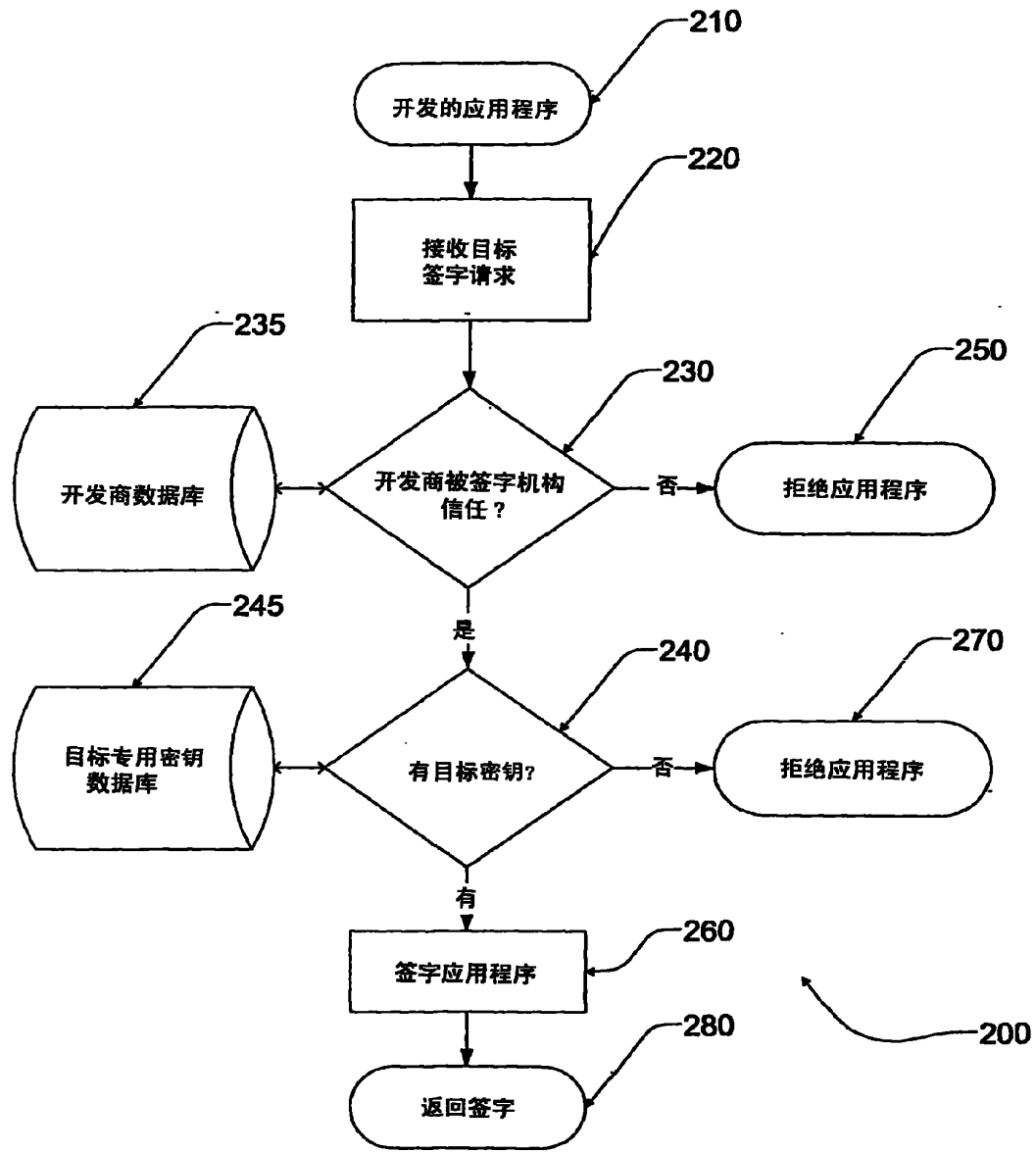


图 5

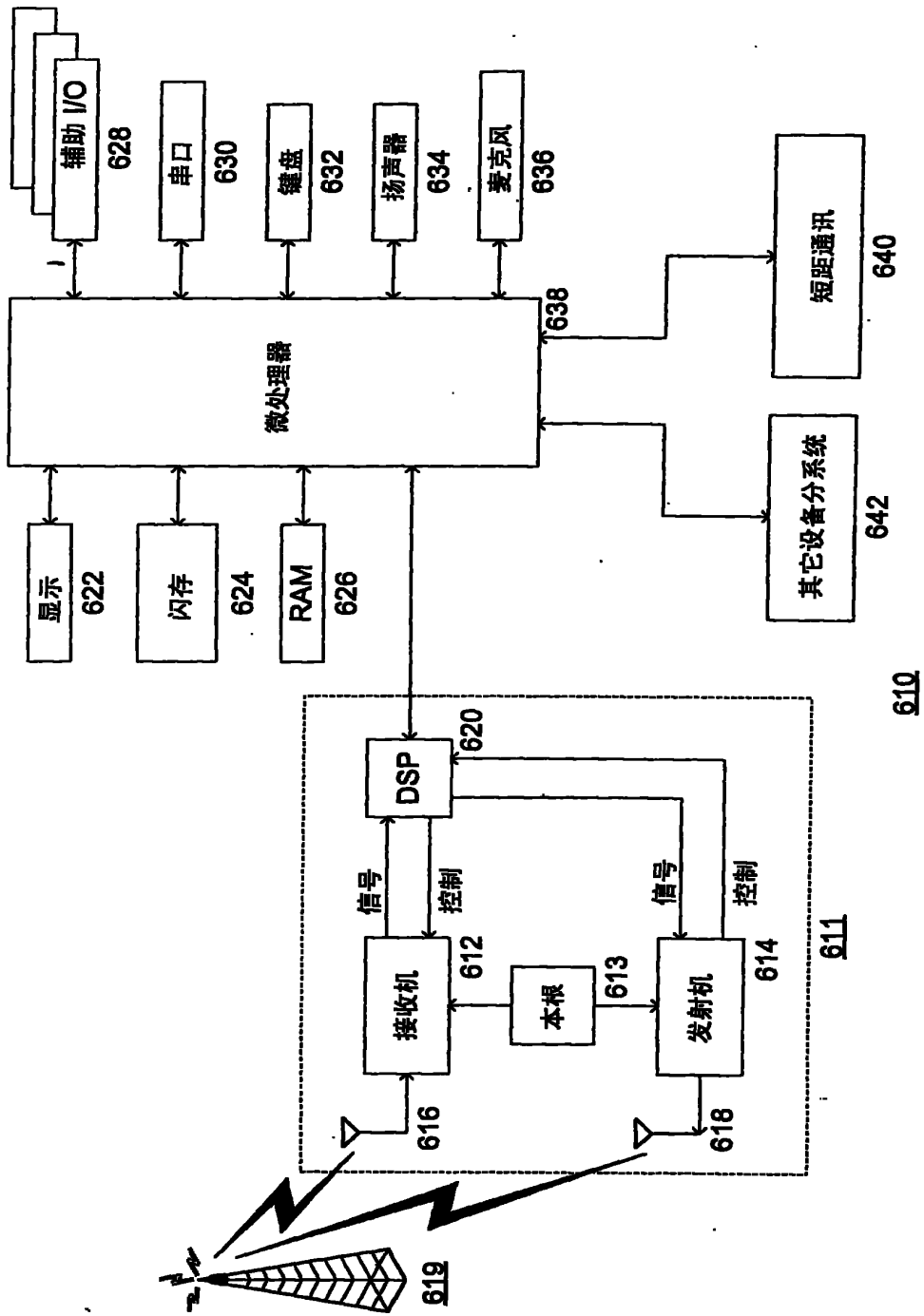


图 6



(12) 发明专利申请

(10) 申请公布号 CN 101694688 A

(43) 申请公布日 2010.04.14

(21) 申请号 200910207912.5

(22) 申请日 2001.09.20

(30) 优先权数据

60/234,152 2000.09.21 US

60/235,354 2000.09.26 US

60/270,663 2001.02.20 US

(62) 分案原申请数据

01819200.9 2001.09.20

(71) 申请人 捷讯研究有限公司

地址 加拿大安大略省沃特卢市

(72) 发明人 戴维·P·亚切 迈克尔斯·S·布朗

赫伯特·A·利特尔

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 戎志敏

(51) Int. Cl.

G06F 21/22 (2006.01)

H04L 29/06 (2006.01)

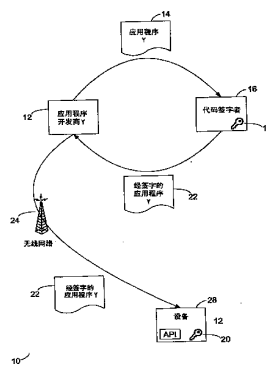
权利要求书 3 页 说明书 14 页 附图 7 页

(54) 发明名称

代码签字系统和方法

(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。



CN 101694688 A

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。

2. 根据权利要求 1 所述的代码签字系统,其特征在于:

(i) 如果数字签字不真实,则虚拟机拒绝软件应用程序访问 API;

或

(ii) 如果数字签字不真实,则虚拟机删除软件应用程序。

3. 根据权利要求 2 所述的代码签字系统,其特征在于:

(iii) 代码签字系统装在移动设备上;

或

(iv) 数字签字由代码签字授权机构产生。

4. 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。

5. 根据权利要求 1 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。

6. 根据权利要求 1 所述的代码签字系统,其特征在于使用专用签字密钥产生数字签字,虚拟机使用公用签字密钥验证数字签字的真实性。

7. 根据权利要求 6 所述的代码签字系统,其特征在于:

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;

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。

8. 根据权利要求 3 所述的系统,其特征在于 API 还包括:

当软件应用程序试图访问敏感的 API 时,移动设备显示描述字符串。

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;

或

一个或多个移动设备的核心功能;

或

移动设备上的硬件。

10. 根据权利要求 9 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。

11. 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。

12. 根据权利要求 1 所述的代码签字系统,其特征在于:

(i) API 与应用平台上的加密流程接口;

(ii) API 与应用平台上的专用数据模块接口。

13. 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。

14. 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,包括步骤:

把软件应用程序装载到移动设备上,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。

15. 根据权利要求 14 所述的方法,其特征还在于还包括:

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。

16. 根据权利要求 14 所述的方法,其特征还在于数字签字和签字标识由代码签字授权机构产生。

17. 根据权利要求 14 所述的方法,其特征还在于还包括步骤:

如果签字标识对应签字标识符,则验证数字签字的真实性;

如果数字签字不真实,则拒绝软件应用程序访问敏感的 API。

18. 根据权利要求 17 所述的方法,其特征还在于还包括步骤:

如果数字签字不真实,则从移动设备上删除软件应用程序。

19. 根据权利要求 17 所述的方法,其特征还在于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:

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,以获得产生的杂乱信号;

将公用签字密钥应用到数字签字,以获得恢复的杂乱信号;

比较产生的杂乱信号与恢复的杂乱信号。

20. 根据权利要求 19 所述的方法,其特征还在于通过计算软件应用程序的杂乱信号和应用专用签字密钥产生数字签字。

21. 根据权利要求 14 所述的方法,其特征还在于还包括:

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。

22. 根据权利要求 21 所述的方法,其特征还在于还包括步骤:

从用户接收指令,准许或拒绝软件应用程序访问敏感的 API。

23. 一种移动设备,包括:

应用平台,具有应用编程接口 API;

验证系统,用于认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 ;
控制系统,允许软件应用程序访问至少一个 API,其中,由软件应用程序提供的数字签字由验证认证 ;

其中,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,以致用于软件应用程序的数字签字根据签字标识的签字方案产生,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允许在多个移动设备的子集上访问。

24. 根据权利要求 23 所述的移动设备,其特征在于还包括验证系统和控制系统的虚拟机,所述虚拟机使 Java 虚拟机,软件应用程序是 Java 应用程序。

25. 根据权利要求 23 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。

26. 根据权利要求 23 所述的移动设备,其特征在于应用平台的 API 访问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。

27. 根据权利要求 23 所述的移动设备,其特征在于使用签字方案下的专用签字密钥产生数字签字,验证系统使用公用签字密钥认证数字签字。

28. 根据权利要求 27 所述的移动设备,其特征在于 :

通过将专用签字密钥应用到签字方案下的软件应用程序的杂乱信号产生数字签字 ;

验证系统通过产生软件应用程序的杂乱信号获得产生的杂乱信号、将公用签字密钥应用到数字签字获得恢复的杂乱信号、验证产生的杂乱信号与恢复的杂乱信号相同来认证数字签字。

29. 根据权利要求 23 所述的移动设备,其特征在于至少一个 API 还包括 :

当软件应用程序试图访问至少一个 API 时,描述字符串被显示给用户。

代码签字系统及方法

[0001] 有关申请的参照

[0002] 本申请要求下列申请的优先权：

[0003] “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

[0004] 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™ 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

[0005] 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者的软件应用程序。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。

[0006] 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。

[0007] 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。

发明内容

[0008] 本发明的目的是提供代码签字系统和方法。

[0009] 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：

[0010] 应用平台；

[0011] 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；

[0012] 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，

[0013] 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。

- [0014] 优选地,如果数字签字不真实,则虚拟机拒绝软件应用程序访问 API。
- [0015] 优选地,如果数字签字不真实,则虚拟机删除软件应用程序。
- [0016] 优选地,代码签字系统装在移动设备上。
- [0017] 优选地,数字签字由代码签字授权机构产生。
- [0018] 优选地,还包括:
- [0019] 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。
- [0020] 优选地,至少一个 API 程序库被分类为敏感的;
- [0021] 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;
- [0022] 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,用于访问敏感的 API 程序库;
- [0023] 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。
- [0024] 优选地,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。
- [0025] 优选地,应用平台包括操作系统。
- [0026] 优选地,包括一个或多个移动设备的核心功能。
- [0027] 优选地,包括移动设备上的硬件。
- [0028] 优选地,硬件包括用户身份模块卡。
- [0029] 优选地,软件应用程序是用于移动设备的 Java 应用程序。
- [0030] 优选地,API 与应用平台上的加密流程接口。
- [0031] 优选地,API 与应用平台上的专用数据模块接口。
- [0032] 优选地,虚拟机是安装在移动设备上的 Java 虚拟机。
- [0033] 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:
 - [0034] 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;
 - [0035] 确定软件应用程序是否包括数字签字和签字标识;
 - [0036] 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;
 - [0037] 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,
 - [0038] 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:
 - [0039] 在移动设备上存储对应专用签字密钥的公用签字密钥;
 - [0040] 产生软件应用程序的杂乱信号来获得产生的杂乱信号;
 - [0041] 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;

- [0042] 比较产生的杂乱信号和恢复的杂乱信号。
- [0043] 优选地,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。
- [0044] 优选地,数字签字和签字标识由代码签字授权机构产生。
- [0045] 优选地,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。
- [0046] 优选地,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。
- [0047] 优选地,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。
- [0048] 优选地,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。
- [0049] 优选地,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。
- [0050] 按照本发明的另一方面,一种移动设备,包括:
- [0051] 应用平台,具有应用编程接口 API;
- [0052] 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;
- [0053] 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;
- [0054] 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,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设备;
- [0055] 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;
- [0056] 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;
- [0057] 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;
- [0058] 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,
- [0059] 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。
- [0060] 优选地,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。
- [0061] 优选地,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。
- [0062] 优选地,应用平台的 API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0063] 优选地,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。

- [0064] 优选地,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。
- [0065] 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:
- [0066] 从软件开发商接收软件应用程序;
- [0067] 确定软件应用程序是否满足至少一个标准;
- [0068] 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;
- [0069] 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;
- [0070] 如果数字签字是真实的,向软件应用程序提供到 API 的路径;
- [0071] 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:
- [0072] 计算软件应用程序的杂乱信号;
- [0073] 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;
- [0074] 在移动设备上提供公用签字密钥;
- [0075] 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;
- [0076] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0077] 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。
- [0078] 优选地,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。
- [0079] 优选地,使用安全的杂乱信号算法计算软件应用程序的杂乱信号。
- [0080] 优选地,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。
- [0081] 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程序编程接口 API 的方法,包括步骤:
- [0082] 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;
- [0083] 接收软件应用程序的杂乱信号;
- [0084] 确定杂乱信号是否是注册的软件开发商所发送;
- [0085] 产生数字签字,其中,
- [0086] 数字签字和签字标识被添加到软件应用程序;
- [0087] 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;
- [0088] 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;
- [0089] 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:
- [0090] 在移动设备上提供公用签字密钥;
- [0091] 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;
- [0092] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0093] 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;

- [0094] 如果数字签字不真实,则拒绝软件应用程序访问敏感的 API。
- [0095] 优选地,产生数字签字的步骤由代码签字授权机构执行。
- [0096] 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:
- [0097] 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;
- [0098] 如果签字标识对应签字标识符,则验证数字签字;
- [0099] 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;
- [0100] 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:
- [0101] 验证与签字标识符对应的签字标识;
- [0102] 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;
- [0103] 产生软件应用程序的杂乱信号,以获得产生的杂乱信号;
- [0104] 把公用签字密钥应用到数字签字,以获得恢复的杂乱信号;
- [0105] 将产生的杂乱信号与恢复的杂乱信号进行比较。
- [0106] 优选地,数字签字和签字标识与移动设备的类型有关。
- [0107] 优选地,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。
- [0108] 优选地,软件应用程序包括多个数字签字和签字标识;
- [0109] 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。
- [0110] 优选地,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。
- [0111] 优选地,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。
- [0112] 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,软件应用程序具有数字签字和签字标识,包括:
- [0113] 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;
- [0114] 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;
- [0115] 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;
- [0116] 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;
- [0117] 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。
- [0118] 优选地,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。
- [0119] 优选地,API 至少接入执行加密算法的加密模块、数据存储器和专用数据模型和用户接口之一。
- [0120] 优选地,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。

[0121] 优选地,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。

[0122] 优选地,要求软件应用程序提供全局数字签字的验证,以访问任何 API。

附图说明

[0123] 图 1 是根据本发明实施例的代码签字协议图;

[0124] 图 2 是图 1 的代码签字协议的流程图;

[0125] 图 3 是在移动设备上的代码签字系统方框图;

[0126] 图 3A 是在一组移动设备上的代码签字系统方框图;

[0127] 图 4 是图 3 和图 3A 代码签字系统的工作流程图;

[0128] 图 5 是管理图 3A 的代码签字真实性的流程图;

[0129] 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

[0130] 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 Java 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。

[0131] 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(如地址簿或日历本)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。

[0132] 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雇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。

[0133] 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。

[0134] 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)。

[0135] 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,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 SHA1)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。

[0136] 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。

[0137] 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。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。

[0138] 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。

[0139] 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。

[0140] 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附加到软件应用程序 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。

[0141] 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。

[0142] 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,或分配到公司雇员。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。

[0143] 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。

[0144] 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。

[0145] 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,并能使虚拟机让数字签字 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。

[0146] 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 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。

[0147] 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 程序库 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,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。

[0148] 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。

[0149] 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。

[0150] 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)。

[0151] 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。

[0152] 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。

[0153] 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。

[0154] 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。

[0155] 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。

[0156] 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)。

[0157] 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,本地振荡器(L0)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™ 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。

[0158] 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。

[0159] 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。

[0160] 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。

[0161] 设备 610 优先包括微处理器 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。

[0162] 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 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。

[0163] 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。

[0164] 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。

[0165] 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。

[0166] 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。

[0167] 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。

[0168] 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 Bluetooth™(蓝牙)通信模式,以提供与有相同能力的系统和设备通信。

[0169] 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。

[0170] 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。

[0171] 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。

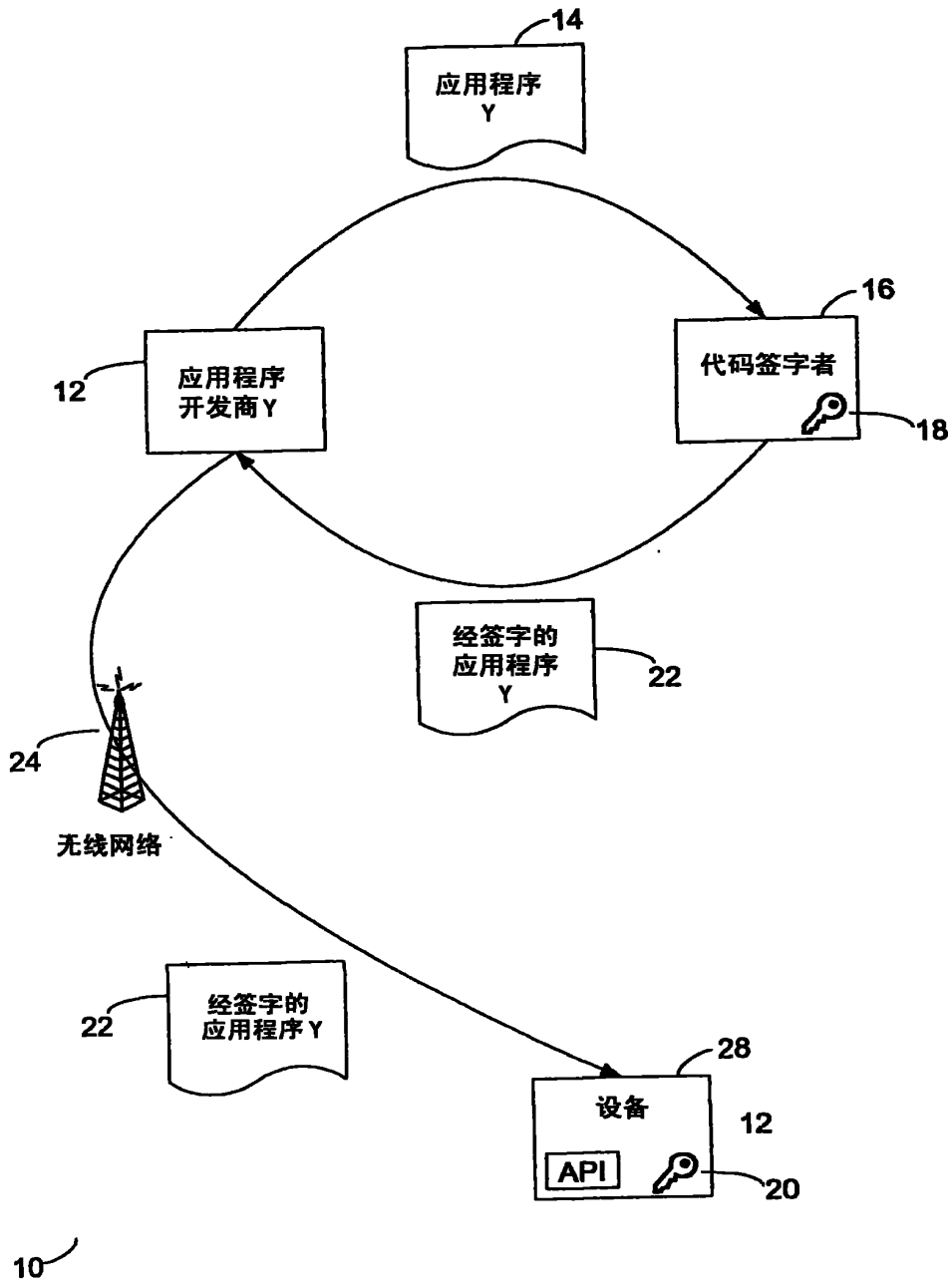


图 1

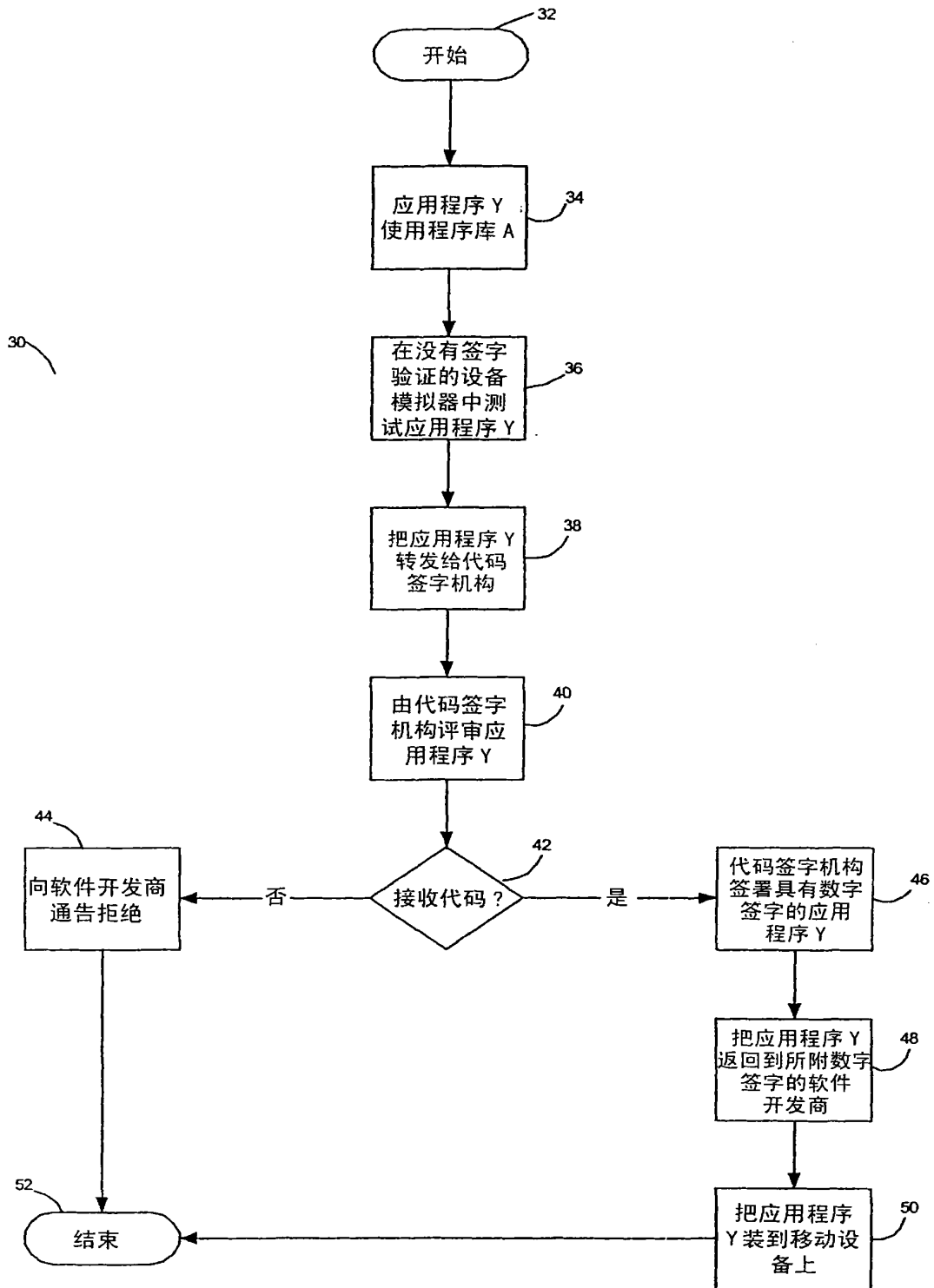


图 2

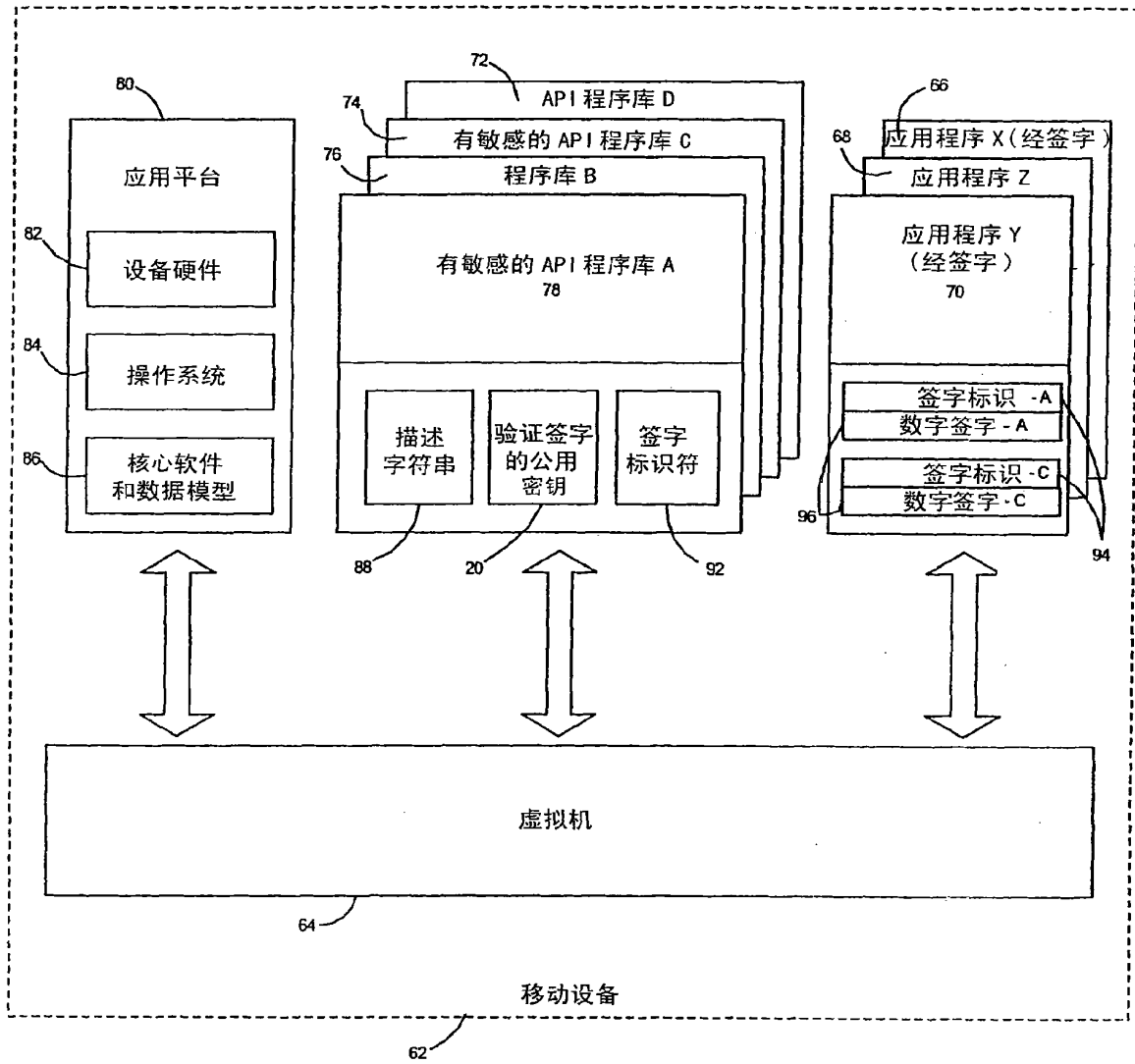


图 3

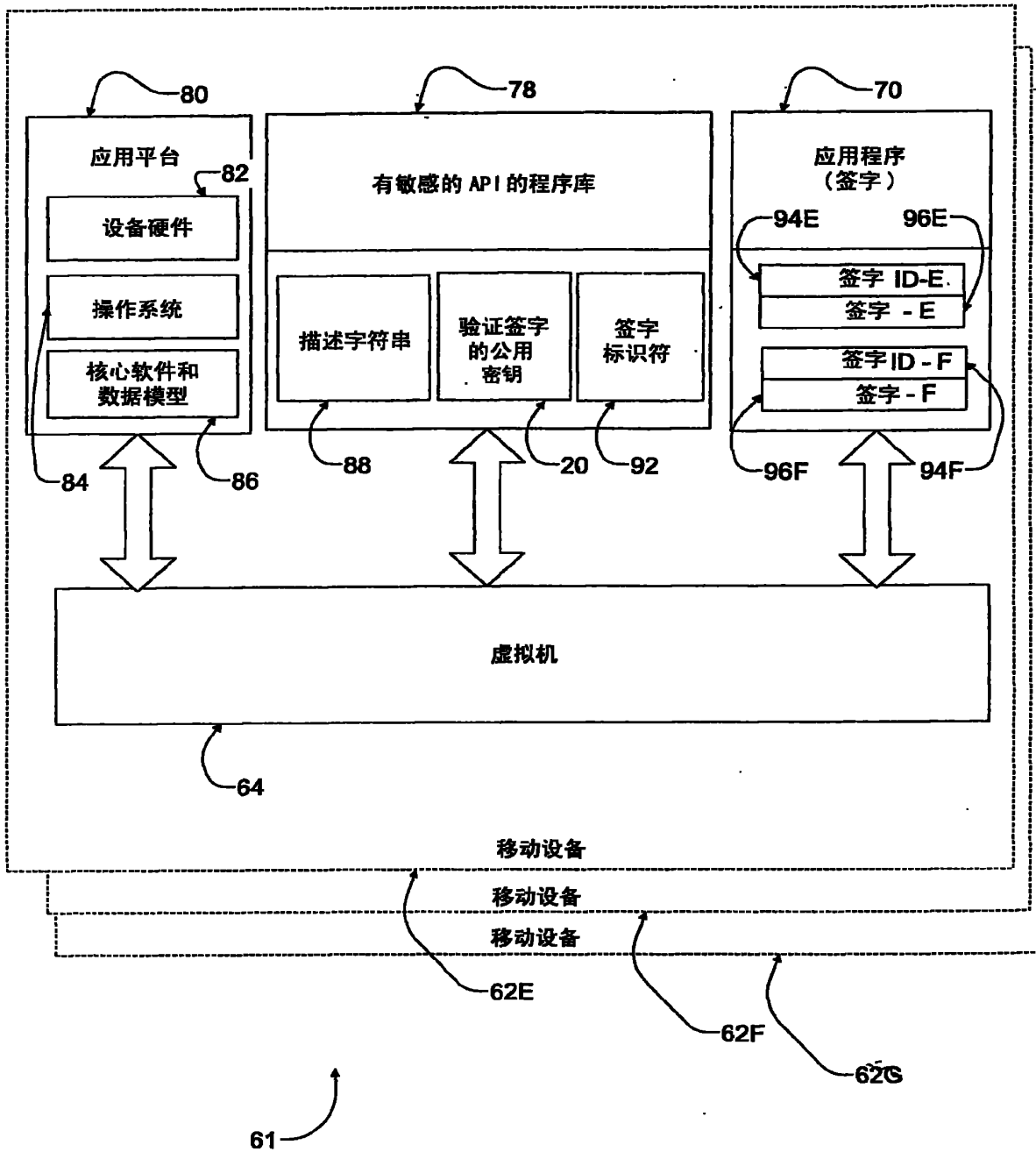


图 3A

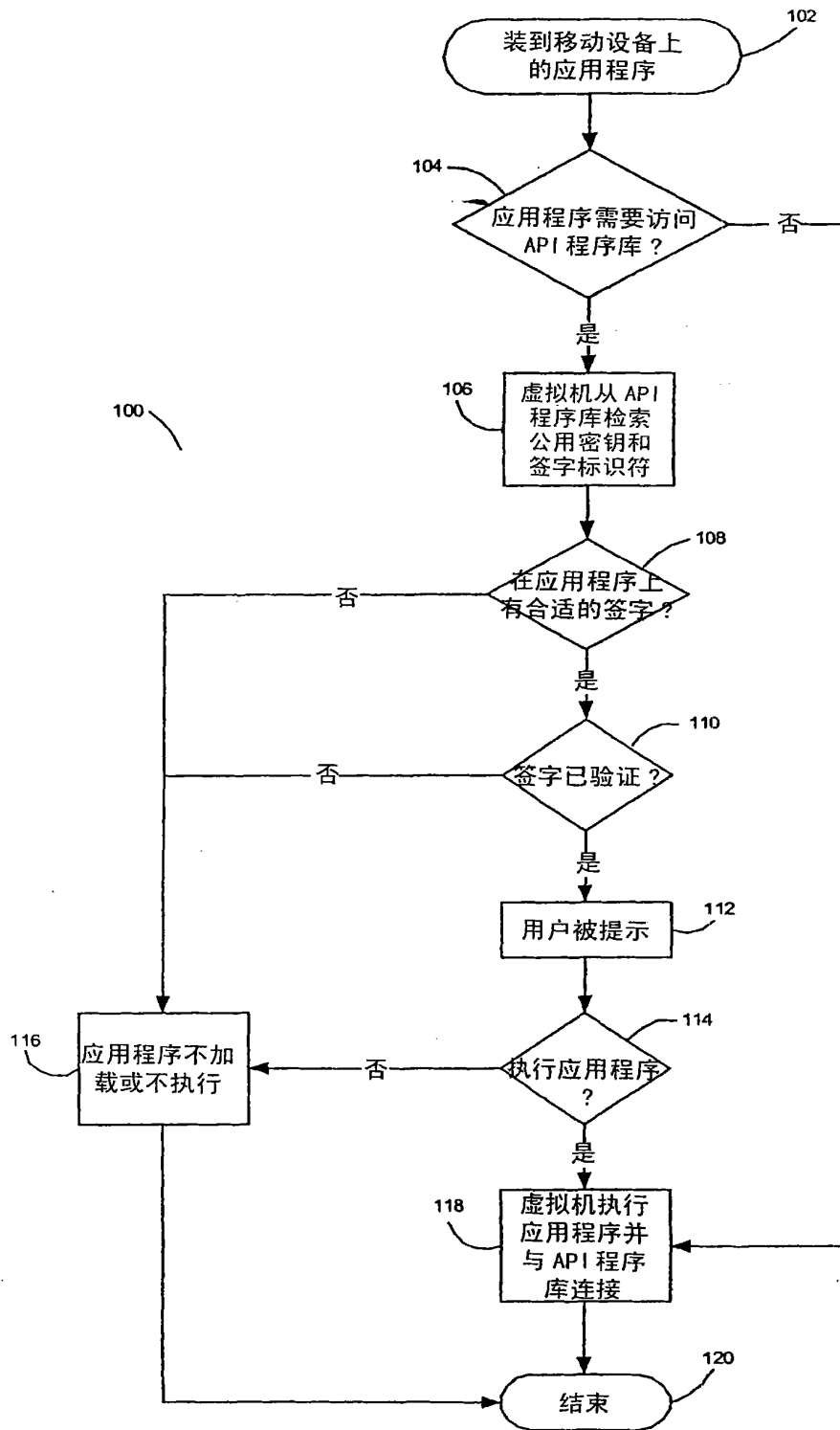


图 4

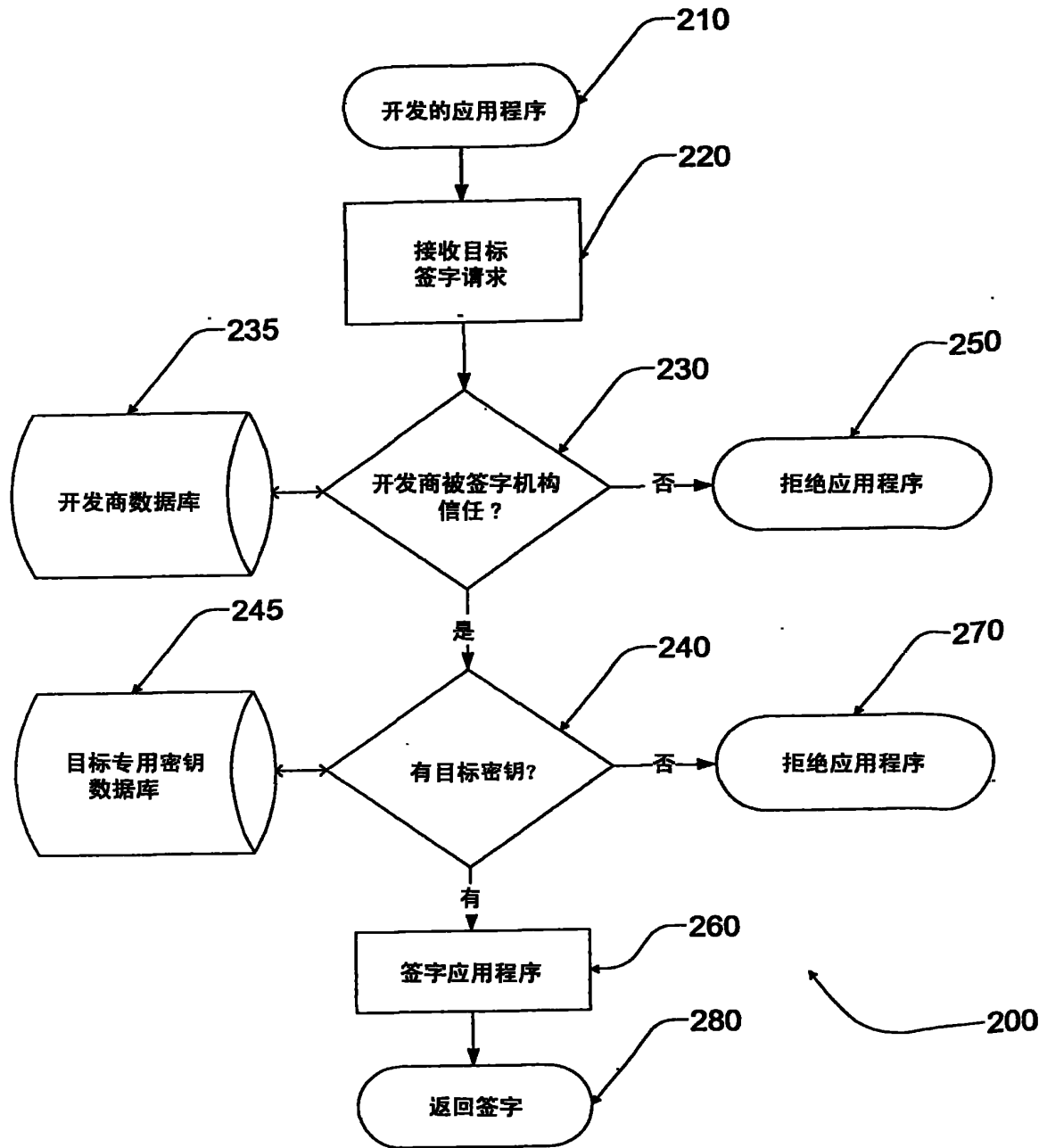


图 5

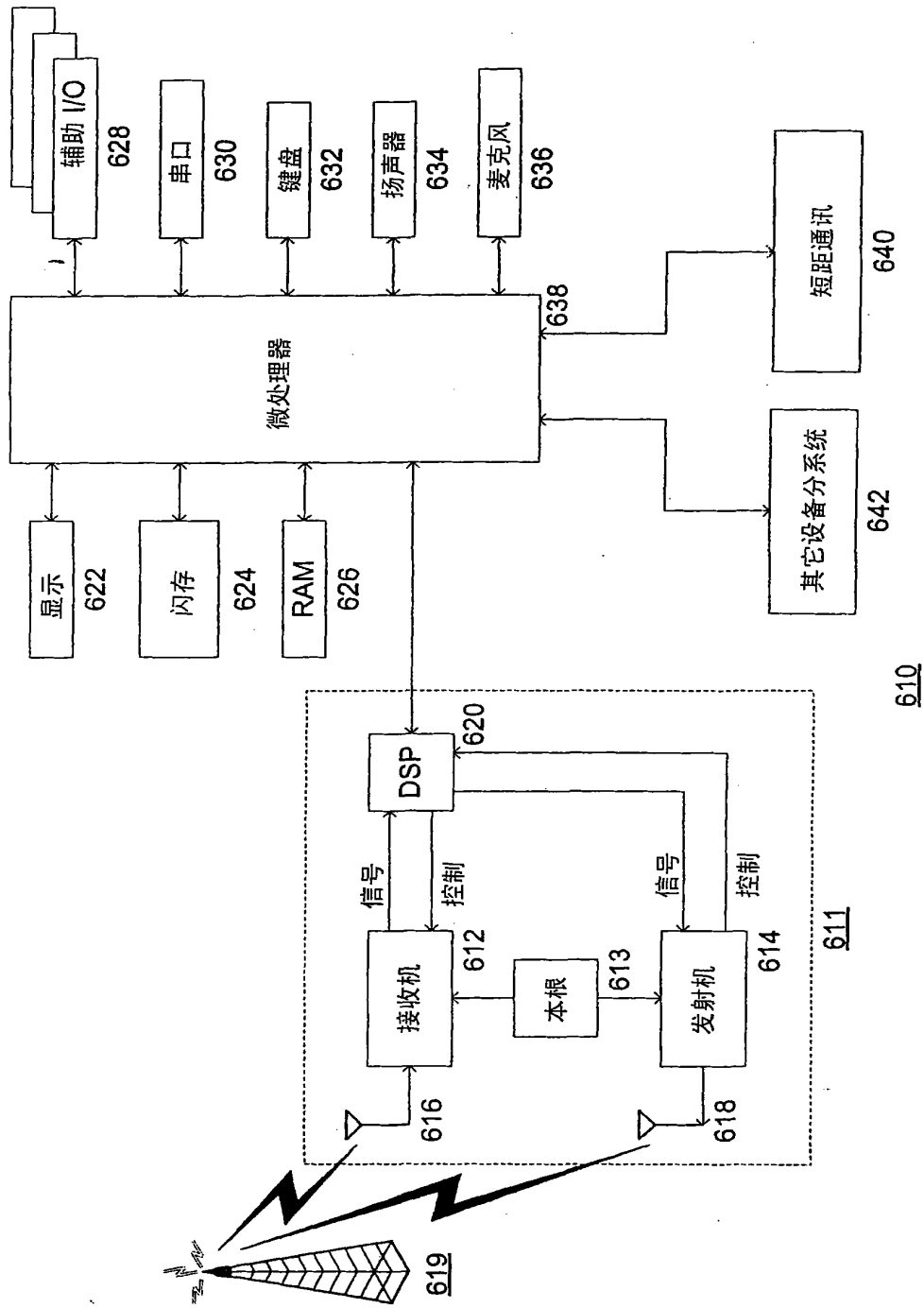


图 6

(19)



(11)

EP 1 320 795 B2

(12)

NEW EUROPEAN PATENT SPECIFICATION

After opposition procedure

(45) Date of publication and mention of the opposition decision:
26.08.2009 Bulletin 2009/35

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

(45) Mention of the grant of the patent:
16.11.2005 Bulletin 2005/46

(86) International application number:
PCT/CA2001/001344

(21) Application number: **01973901.0**

(87) International publication number:
WO 2002/025409 (28.03.2002 Gazette 2002/12)

(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

• **LITTLE, Herbert, A.**
Waterloo, Ontario N2T 2V8 (CA)

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

(74) Representative: **Dagg, Nicola Helen**
Allen & Overy LLP
One Bishops Square
London E1 6AO (GB)

(43) Date of publication of application:
25.06.2003 Bulletin 2003/26

(56) References cited:
EP-A- 0 930 793 WO-A-99/05600
US-A- 5 978 484 US-A- 6 157 721

(60) Divisional application:
05024661.0 / 1 626 324
05024662.8 / 1 626 325
05024663.6 / 1 626 326

- " "Additional interindustry commands and security attributes" ISO/IEC FCD 7816-9 PART 9
- " "Security related interindustry commands" ISO/IEC FDIS 7816-8 PART 8
- " "Interindustry commands for interchange" ISO/IEC 7816-4 PART 4 pages 12 - 16
- **RANKL, WOLFGANG UND EFFING, WOLFGANG:** 'Handbuch der Chipkarten', 3. AUFLAGE, 1999, ISBN 3-446-21115-2 article 'pp. 197-203, 261-272, 740, 795-797'
- " "Interindustry commands for interchange" ISO/IEC 7816-4 PART 4 XP002269400

(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)

EP 1 320 795 B2

DescriptionBACKGROUND1. 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™ 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 US 5,978,484 illustrates a system for distribution of digitally signed executable objects using an RSA encrypted SHA-hash.

[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 au-

thorities 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 soft-

ware 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 applica-

tions, 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™ (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 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 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 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 man-

agement 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 mul-

tiple 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™ mobile communication system or DataTAC™ 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 software applications as described above, may also be loaded onto the device 610 through the network 619, an auxiliary I/O subsystem 628, 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™ communication module to provide for communication with similarly-enabled systems and devices.

[0052] The embodiments described herein are exam-

ples 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.

Claims

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 is associated with the signature identification, comprising:

- an application platform;
- an application programming interface (API) having an associated signature identifier (92), the API is configured to link the software application with the application platform; and
- wherein the authenticity of the digital signature is verified by a virtual machine (69) in order to control access to the API by the software application where the signature identifier corresponds to the signature identification;
- wherein the digital signature is generated by applying a 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 a public signature key to the signature to obtain a recovered hash, and comparing the generated hash with the recovered hash.

2. The code signing system of claim 1, wherein

- (i) the virtual machine denies the software application access to the API if the digital signature is not authenticated, or
- (ii) wherein the virtual machine purges the software application if the digital signature is not authenticated,

3. The code signing system of claim 1 or 2, wherein

- (iii) the code signing system is installed on a mobile device, or
- (iv) wherein the digital signature is generated by a code signing authority.

4. The code signing system of any of claims 1 to 3, 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.

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

- 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.
6. The code signing system of claim 3, wherein the API further comprises:
- a description string that is displayed by the mobile device when the software application attempts to access the API.
7. The code signing system of any of claims 1 to 6, wherein the application platform
- (i) comprises an operating system, or
 - (ii) comprises one or more core functions of a mobile device, or
 - (iii) comprises hardware on a mobile device.
8. The code signing system of claim 7, wherein the hardware comprises a subscriber identity module (SIM) card.
9. The code signing system of any of claims 1 to 8, wherein the software application is a Java application for a mobile device.
10. The code signing system of any of claims 1 to 9, wherein
- (i) the API interfaces with a cryptographic routine on the application platform, or wherein
 - (ii) the API interfaces with a proprietary data model on the application platform.
11. The code signing system of any of claims 1 to 10, wherein the virtual machine is a Java virtual machine installed on a mobile device.
12. 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 that requires access to a sensitive application programming interface (API) having a signature identifier (92);
 - determining whether the software application includes a digital signature (96) and a signature identification (94);
 - denying the software application access to the sensitive API where the digital signature does not correspond with the signature identifier;
 - verifying the authenticity of the digital signature where the signature identification corresponds with the signature identifier, wherein access to the sensitive API by the software application is based upon the verifying of the authenticity of the digital signature;
 - 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 steps comprising:
 - 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.
13. The method of claim 12, comprising the additional step of:
- purging the software application from the mobile device where the signature identification does not correspond with the signature identifier.
14. The method of claim 12 or claim 13, wherein the digital signature and the signature identification are generated by a code signing authority.
15. The method of any of claims 12 to 14, comprising the additional step of:
- denying the software application access to the sensitive API where the digital signature is not authenticated.
16. The method of claim 15, comprising the additional step of:
- purging the software application from the mobile device where the digital signature is not authenticated.
17. The method according to one of the claims 12 to 16, wherein a description string is displayed to a user when the software application attempts to access said at least one of the APIs.
18. The method of any of claims 12 to 17, 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.

19. The method of claim 18, comprising the additional step of:

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

Patentansprüche

1. Code-signierendes System zum Betreiben in Verbindung mit einer Softwareanwendung (66), welche eine digitale Signatur (96) und einen Signaturidentifikator (94) aufweist, wobei die digitale Signatur dem Signaturidentifikator zugeordnet ist, welches umfasst:

- eine Anwendungsplattform;
- eine Schnittstelle für das Anwendungsprogramm (API), welche eine zugeordnete Signaturkennung (92) aufweist, wobei besagte Schnittstelle (API) so konfiguriert ist, dass sie die Softwareanwendung mit der Anwendungsplattform verbindet; und
- wobei die Echtheit der digitalen Signatur durch eine virtuelle Maschine (69) überprüft wird, um den Zugang zur API durch die Softwareanwendung zu kontrollieren, wenn die Signaturkennung dem Signaturidentifikator entspricht;
- wobei die digitale Signatur durch Anwenden eines privaten Signaturschlüssels auf einen Hash-Wert der Softwareanwendung erzeugt wird, und die virtuelle Maschine die Echtheit der digitalen Signatur überprüft durch Erzeugen eines Hash-Werts der Softwareanwendung, um einen erzeugten Hash-Wert zu erhalten, Anwenden eines öffentlichen Signaturschlüssels auf die Signatur, um einen wiederhergestellten Hash-Wert zu erhalten, und Vergleichen des erzeugten Hash-Werts mit dem wiederhergestellten Hash-Wert.

2. Code-signierendes System nach Anspruch 1, wobei

- (i) die virtuelle Maschine der Softwareanwendung den Zugang zur API verweigert, falls die digitale Signatur nicht bestätigt wird, oder
- (ii) wobei die virtuelle Maschine die Softwareanwendung entfernt, wenn die digitale Signatur nicht bestätigt wird.

3. Code-signierendes System nach Anspruch 1 oder 2, wobei

- (iii) das Code-signierende System in einer mobilen Vorrichtung installiert ist, oder
- (iv) wobei die digitale Signatur durch eine Code-signierende Autorität erzeugt wird.

4. Code-signierendes System nach einem der Ansprüche 1 bis 3, welches weiter umfasst:

- eine Vielzahl von API-Bibliotheken, wobei jede API-Bibliothek aus der Vielzahl der API-Bibliotheken eine Vielzahl von APIs enthält, wobei die virtuelle Maschine den Zugang zur Vielzahl der API-Bibliotheken durch die Softwareanwendung regelt.

5. Code-signierendes System nach einem der Ansprüche 1 bis 4,

- wobei wenigstens eine aus der Vielzahl der API-Bibliotheken als sensibel eingeordnet wird;
- wobei der Zugang zu einer sensiblen API-Bibliothek eine digitale Signatur erfordert, welche einem Signaturidentifikator zugeordnet ist, wobei der Signaturidentifikator einer Signaturkennung entspricht, die der sensiblen API-Bibliothek zugeordnet ist;
- wobei die Softwareanwendung wenigstens eine digitale Signatur und wenigstens einen zugeordneten Signaturidentifikator zum Zugang sensibler API-Bibliotheken enthält; und
- wobei die virtuelle Maschine die Softwareanwendung zum Zugang der sensiblen API-Bibliothek durch Überprüfen der einen digitalen Signatur berechtigt, welche in der Softwareanwendung enthalten ist, die einen Signaturidentifikator aufweist, der der Signaturkennung der sensiblen API-Bibliothek entspricht.

6. Code-signierendes System nach Anspruch 3, wobei die API weiter umfasst:

- eine Beschreibungszeichenfolge, die durch die mobile Vorrichtung angezeigt wird, wenn die Softwareanwendung den Zugang zur API - versucht.

7. Code-signierendes System nach einem der Ansprüche 1 bis 6, wobei die Anwendungsplattform

- (i) ein Betriebssystem umfasst, oder
- (ii) eine oder mehrere Kernfunktionen einer mobilen Vorrichtung umfasst, oder
- (iii) eine Hardware in einer mobilen Vorrichtung umfasst.

8. Code-signierendes System nach Anspruch 7, wobei die Hardware eine Karte für ein Teilnehmeridentifizierungsmodul (SIM) umfasst.

9. Code-signierendes System nach einem der Ansprüche 1 bis 8, wobei die Softwareanwendung eine Java-Anwendung für eine tragbare Vorrichtung ist.

10. Code-signierendes System nach einem der Ansprüche 1 bis 9, wobei
- (i) die API sich an ein Verschlüsselungsprogramm in der Anwendungsplattform anschließen lässt, oder wobei
 - (ii) die API sich an ein proprietäres Datenmodell in der Anwendungsplattform anschließen lässt.
11. Code-signierendes System nach einem der Ansprüche 1 bis 10, wobei die virtuelle Maschine eine virtuelle Java-Maschine ist, die in einer mobilen Vorrichtung installiert ist.
12. Verfahren zur Regelung des Zugangs zu sensiblen Schnittstellen für Anwendungsprogramme in einer mobilen Vorrichtung (62), welches die folgenden Schritte umfasst:
- Laden einer Softwareanwendung (66) in der mobilen Vorrichtung, welche den Zugang zu einer sensiblen Schnittstelle für ein Anwendungsprogramm (API) mit einer Signaturkennung (92) fordert;
 - Bestimmen, ob die Softwareanwendung eine digitale Signatur (96) und einen Signaturidentifikator (94) enthält;
 - Verweigern der Softwareanwendung den Zugang zur sensiblen API, wenn die digitale Signatur nicht der Signaturkennung entspricht;
 - Überprüfen der Echtheit der digitalen Signatur, wenn der Signaturidentifikator der Signaturkennung entspricht, wobei der Zugang zu der sensiblen API durch die Softwareanwendung auf dem Überprüfen der Echtheit der digitalen Signatur basiert;
 - wobei die digitale Signatur durch Anwenden eines privaten Signaturschlüssels auf einen Hash-Wert der Softwareanwendung erzeugt wird, und wobei der Schritt des Überprüfens der Echtheit der digitalen Signatur durch Schritte ausgeführt wird, welche umfassen:
 - Speichern eines öffentlichen Signaturschlüssels, welcher dem privaten Signaturschlüssel in der mobilen Vorrichtung entspricht;
 - Erzeugen eines Hash-Werts der Softwareanwendung, um einen erzeugten Hash-Wert zu erhalten;
 - Anwenden des öffentlichen Signaturschlüssels auf die digitale Signatur, um einen wiederhergestellten Hash-Wert zu erhalten; und
 - Vergleichen des erzeugten Hash-Werts mit dem wiederhergestellten Hash-Wert.
13. Verfahren nach Anspruch 12, welches den folgenden zusätzlichen Schritt enthält:
- Entfernen der Softwareanwendung aus der mobilen Vorrichtung, wenn der Signaturidentifikator nicht der Signaturkennung entspricht.
14. Verfahren nach Anspruch 12 oder 13, wobei die digitale Signatur und der Signaturidentifikator durch eine Code-signierende Autorität erzeugt werden.
15. Verfahren nach einem der Ansprüche 12 bis 14, welches den folgenden zusätzlichen Schritt aufweist:
- Verweigern der Softwareanwendung den Zugang zur sensiblen API, wenn die digitale Signatur nicht bestätigt wird.
16. Verfahren nach Anspruch 15, welches den folgenden zusätzlichen Schritt umfasst:
- Entfernen der Softwareanwendung aus der mobilen Vorrichtung, wenn die digitale Signatur nicht bestätigt wird.
17. Verfahren nach einem der Ansprüche 12 bis 16, wobei eine Beschreibungszeichenfolge einem Benutzer angezeigt wird, wenn die Softwareanwendung den Zugang zu wenigstens einer der APIs versucht.
18. Verfahren nach einem der Ansprüche 12 bis 17, welches den folgenden zusätzlichen Schritt umfasst:
- Anzeigen einer Beschreibungszeichenfolge, welche einem Benutzer der mobilen Vorrichtung anzeigt, dass die Softwareanwendung den Zugang zur sensiblen API fordert.
19. Verfahren nach Anspruch 18, welches den zusätzlichen folgenden Schritt umfasst:
- Empfangen eines Kommandos vom Benutzer, welches der Softwareanwendung Zugang zur sensiblen API gewährt oder verweigert.
- Revendications**
1. Système de signature par code à utiliser avec une application de logiciel (66) comportant une signature numérique (96) et une identification de signature (94), la signature numérique étant associée à l'identification de signature, comprenant :
- une plate-forme d'application ;
 - une interface de programmation d'applications (IPA) présentant un identificateur de signature associé (92), l'IPA étant configurée pour lier l'application de logiciel à la plate-forme

- d'application ; et
- dans lequel l'authenticité de la signature numérique est vérifiée par une machine virtuelle (69) afin de commander l'accès à l'IPA par l'application de logiciel lorsque l'identificateur de signature correspond à l'identification de signature ;
 - dans lequel la signature numérique est générée en appliquant une clé de signature privée à un adressage de l'application de logiciel, et la machine virtuelle vérifie l'authenticité de la signature numérique en générant un adressage de l'application de logiciel afin d'obtenir un adressage généré, en appliquant une clé de signature publique à la signature pour obtenir un adressage récupéré, et en comparant l'adressage généré avec l'adressage récupéré.
2. Système de signature par code selon la revendication 1, dans lequel :
- (i) la machine virtuelle refuse l'accès de l'application de logiciel à l'IPA si la signature numérique n'est pas authentifiée ; ou
 - (ii) la machine virtuelle purge l'application de logiciel si la signature numérique n'est pas authentifiée.
3. Système de signature par code selon la revendication 1 ou 2, dans lequel :
- (iii) le système de signature par code est installé sur un dispositif mobile ; ou
 - (iv) la signature numérique est générée par une autorité de signature par code.
4. Système de signature par code selon l'une quelconque des revendications 1 à 3, comprenant en outre :
- une pluralité de bibliothèques d'IPA, chacune de la pluralité de bibliothèques d'IPA comprenant une pluralité d'IPA, la machine virtuelle commandant l'accès à la pluralité de bibliothèques d'IPA par l'application de logiciel.
5. Système de signature par code selon l'une quelconque des revendications 1 à 4,
- dans lequel au moins une parmi la pluralité de bibliothèques d'IPA est classée comme confidentielle ;
 - dans lequel l'accès à une bibliothèque d'IPA confidentielle nécessite une signature numérique associée à une identification de signature pour laquelle l'identification de signature correspond à un identificateur de signature associé à la bibliothèque d'IPA confidentielle ;
 - dans lequel l'application de logiciel contient au
- moins une signature numérique et au moins une identification de signature associée pour accéder aux bibliothèques d'IPA confidentielles ; et
- dans lequel la machine virtuelle authentifie l'application de logiciel pour accéder à la bibliothèque d'IPA confidentielle en vérifiant la signature numérique incorporée dans l'application de logiciel présentant une identification de signature correspondant à l'identificateur de signature de la bibliothèque d'IPA confidentielle.
6. Système de signature par code selon la revendication 3, dans lequel l'IPA comprend en outre :
- une chaîne de description qui est affichée par le dispositif mobile lorsque l'application de logiciel tente d'accéder à l'IPA.
7. Système de signature par code selon l'une quelconque des revendications 1 à 6, dans lequel la plateforme d'application :
- (i) comprend un système d'exploitation, ou
 - (ii) comprend une ou plusieurs fonction(s) centrale(s) d'un dispositif mobile, ou
 - (iii) comprend du matériel sur un dispositif mobile.
8. Système de signature par code selon la revendication 7, dans lequel le matériel comprend une carte de module d'identité d'abonné (SIM).
9. Système de signature par code selon l'une quelconque des revendications 1 à 8, dans lequel l'application de logiciel est une application Java pour un dispositif mobile.
10. Système de signature par code selon l'une quelconque des revendications 1 à 9, dans lequel :
- (i) l'IPA interface avec une routine cryptographique sur la plate-forme d'application, ou dans lequel :
 - (ii) l'IPA interface avec un modèle de données propriétaire sur la plate-forme d'application.
11. Système de signature par code selon l'une quelconque des revendications 1 à 10, dans lequel la machine virtuelle est une machine virtuelle Java installée sur un dispositif mobile.
12. Procédé pour commander l'accès à des interfaces de programmation d'applications confidentielles sur un dispositif mobile (62), comprenant les étapes consistant à :
- charger une application de logiciel (66) sur le dispositif mobile, qui requiert un accès à une

- interface de programmation d'applications (IPA) confidentielle comprenant un identificateur de signature (92) ;
- déterminer si l'application de logiciel contient une signature numérique (96) et une identification de signature (94) ;
 - refuser l'accès de l'application de logiciel à l'IPA confidentielle si la signature numérique ne correspond pas à l'identificateur de signature ;
 - vérifier l'authenticité de la signature numérique si l'identification de signature correspond à l'identificateur de signature, l'accès à l'IPA confidentielle par l'application de logiciel étant basé sur la vérification de l'authenticité de la signature numérique ;
 - dans lequel la signature numérique est générée en appliquant une clé de signature privée à un adressage de l'application de logiciel, et dans lequel l'étape de vérification de l'authenticité de la signature numérique est exécutée par les étapes consistant à :
 - stocker une clé de signature publique correspondant à la clé de signature privée sur le dispositif mobile ;
 - générer un adressage de l'application de logiciel pour obtenir un adressage généré ;
 - appliquer la clé de signature publique à la signature numérique pour obtenir un adressage récupéré ; et
 - comparer l'adressage généré avec l'adressage récupéré.
13. Procédé selon la revendication 12, comprenant l'étape supplémentaire consistant à :
- purger l'application de logiciel du dispositif mobile lorsque l'identification de signature ne correspond pas à l'identificateur de signature.
14. Procédé selon la revendication 12 ou la revendication 13, dans lequel la signature numérique et l'identification de signature sont générées par une autorité de signature par code.
15. Procédé selon l'une quelconque des revendications 12 à 14, comprenant l'étape supplémentaire consistant à :
- refuser l'accès de l'application de logiciel à l'IPA confidentielle lorsque la signature numérique n'est pas authentifiée.
16. Procédé selon la revendication 15, comprenant l'étape supplémentaire consistant à :
- purger l'application de logiciel du dispositif mobile lorsque la signature numérique n'est pas authentifiée.
17. Procédé selon l'une quelconque des revendications 12 à 16, dans lequel une chaîne de description est affichée pour un utilisateur lorsque l'application de logiciel tente d'accéder à ladite au moins une des IPA.
18. Procédé selon l'une quelconque des revendications 12 à 17, comprenant l'étape supplémentaire consistant à :
- afficher une chaîne de description qui notifie à un utilisateur du dispositif mobile que l'application de logiciel réclame un accès à l'IPA confidentielle.
19. Procédé selon la revendication 18, comprenant l'étape supplémentaire consistant à :
- recevoir une instruction en provenance de l'utilisateur accordant ou refusant l'accès de l'application de logiciel à l'IPA confidentielle.

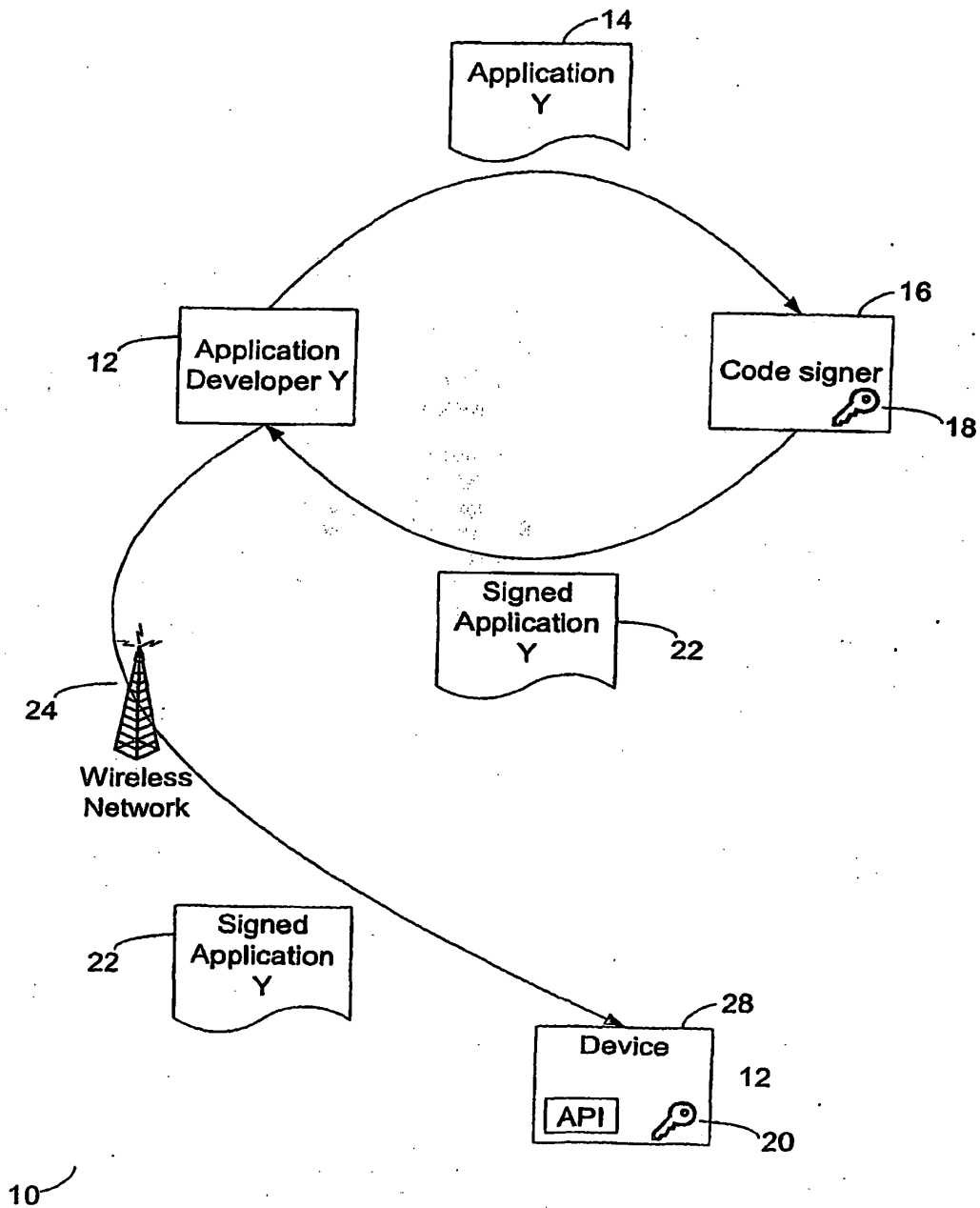
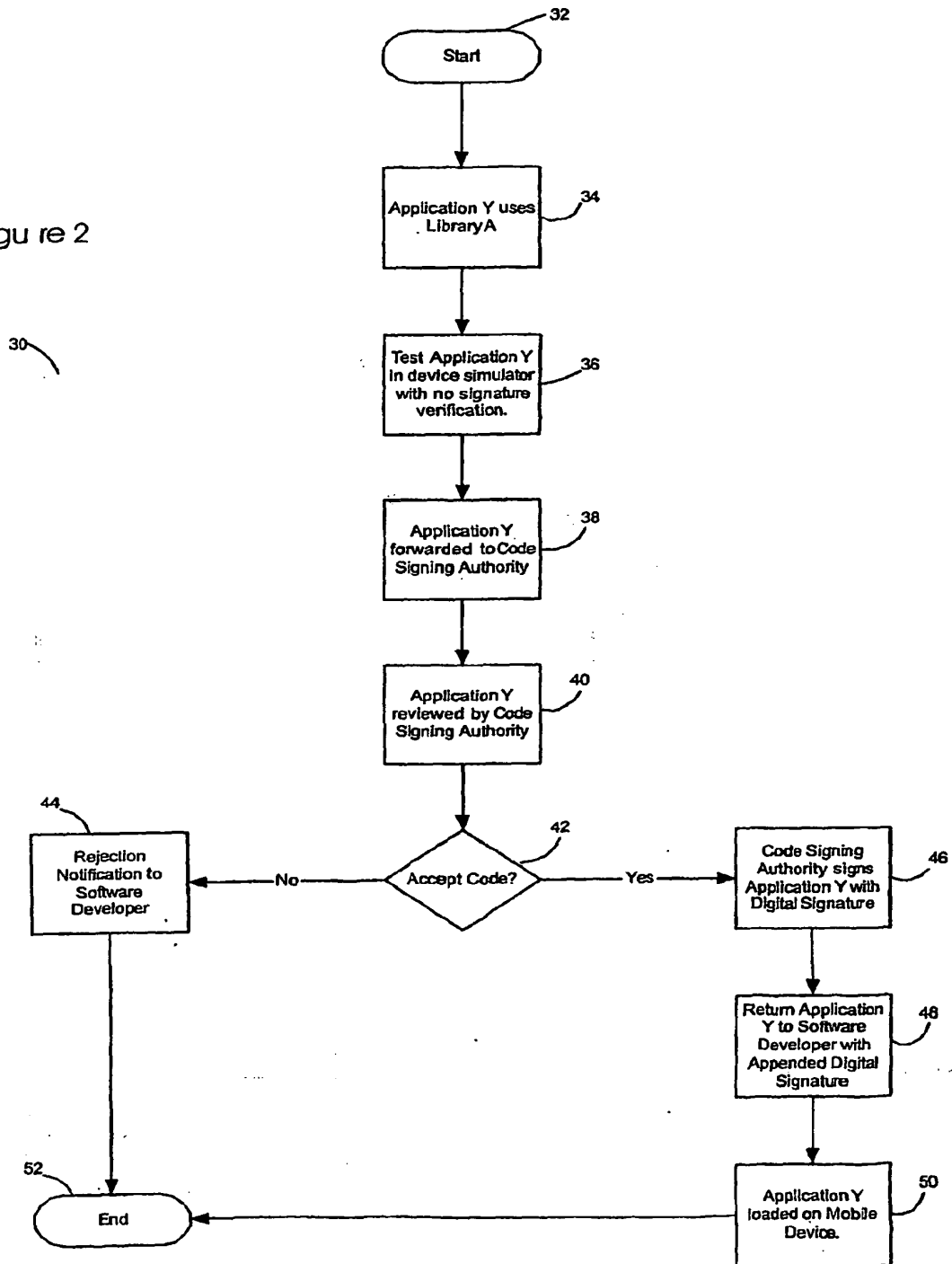
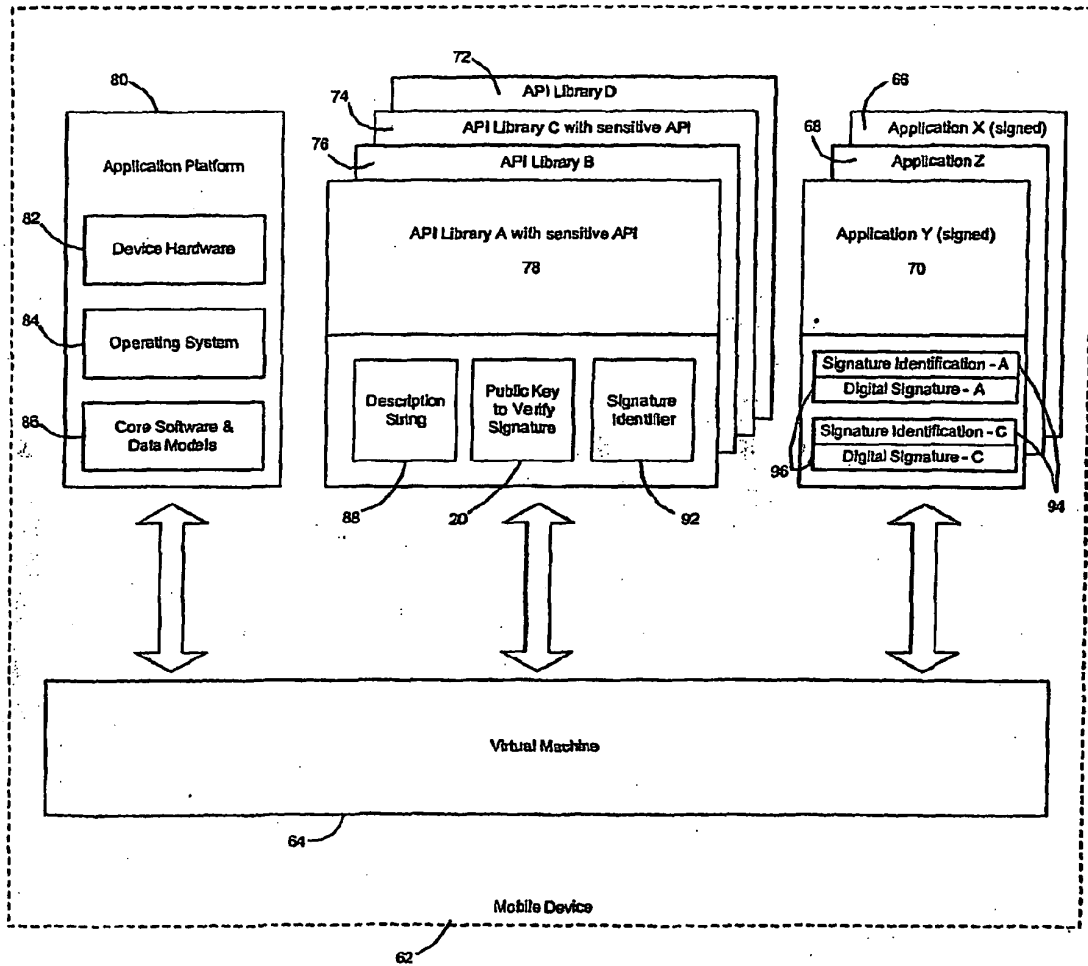


Figure 1

Figure 2





60

Figure 3

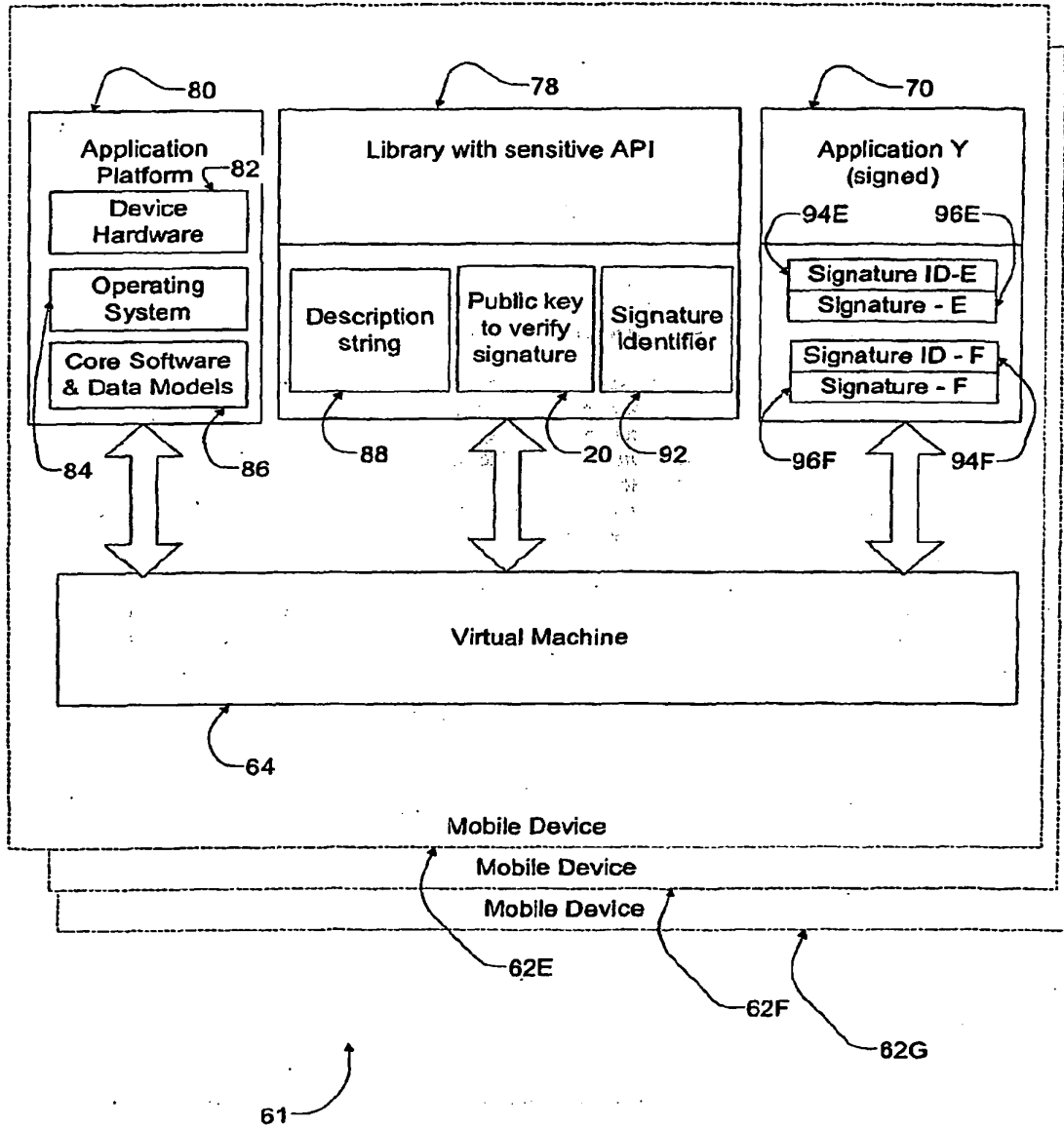
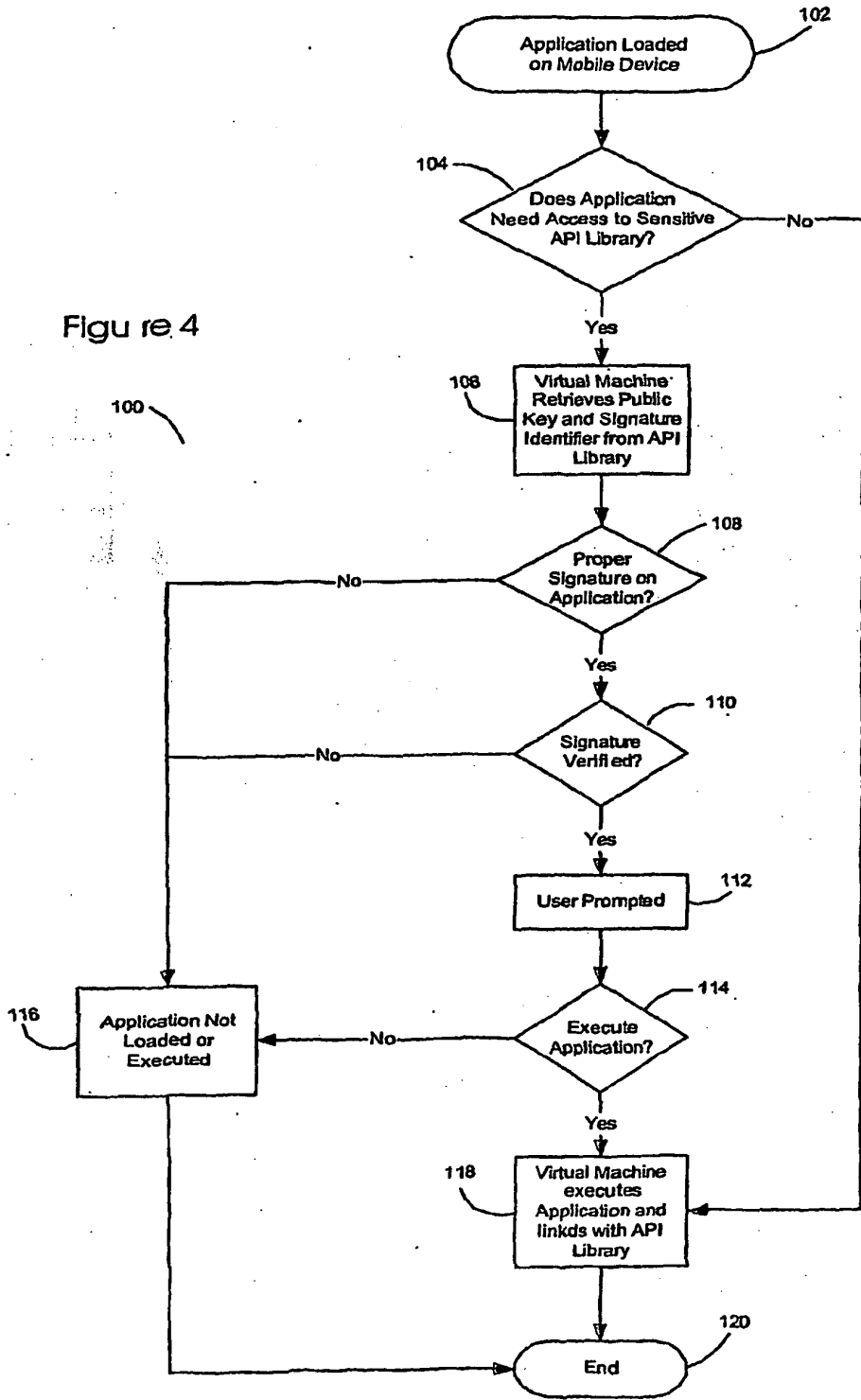


Figure 3A

Figure 4



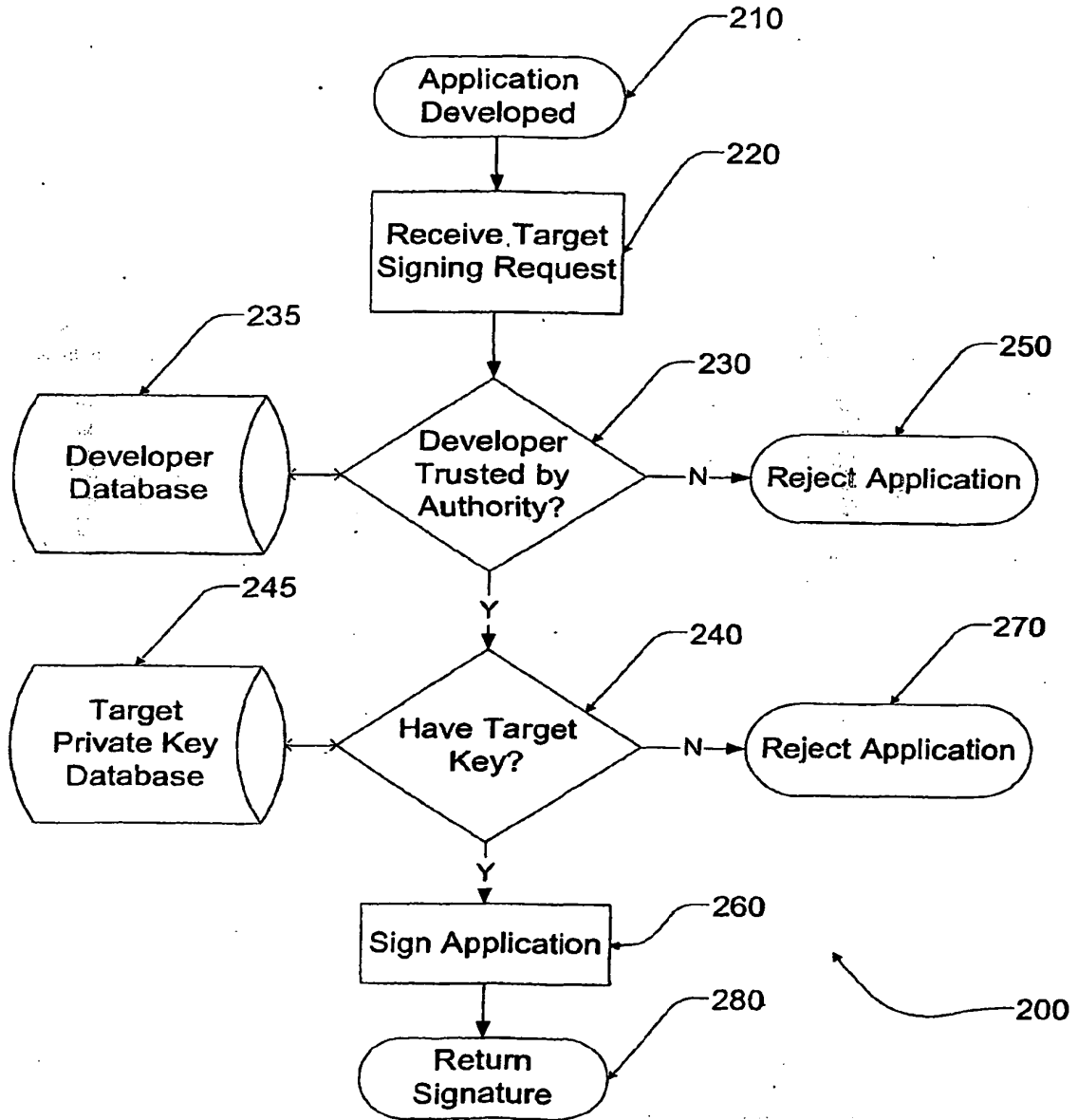


Figure 5

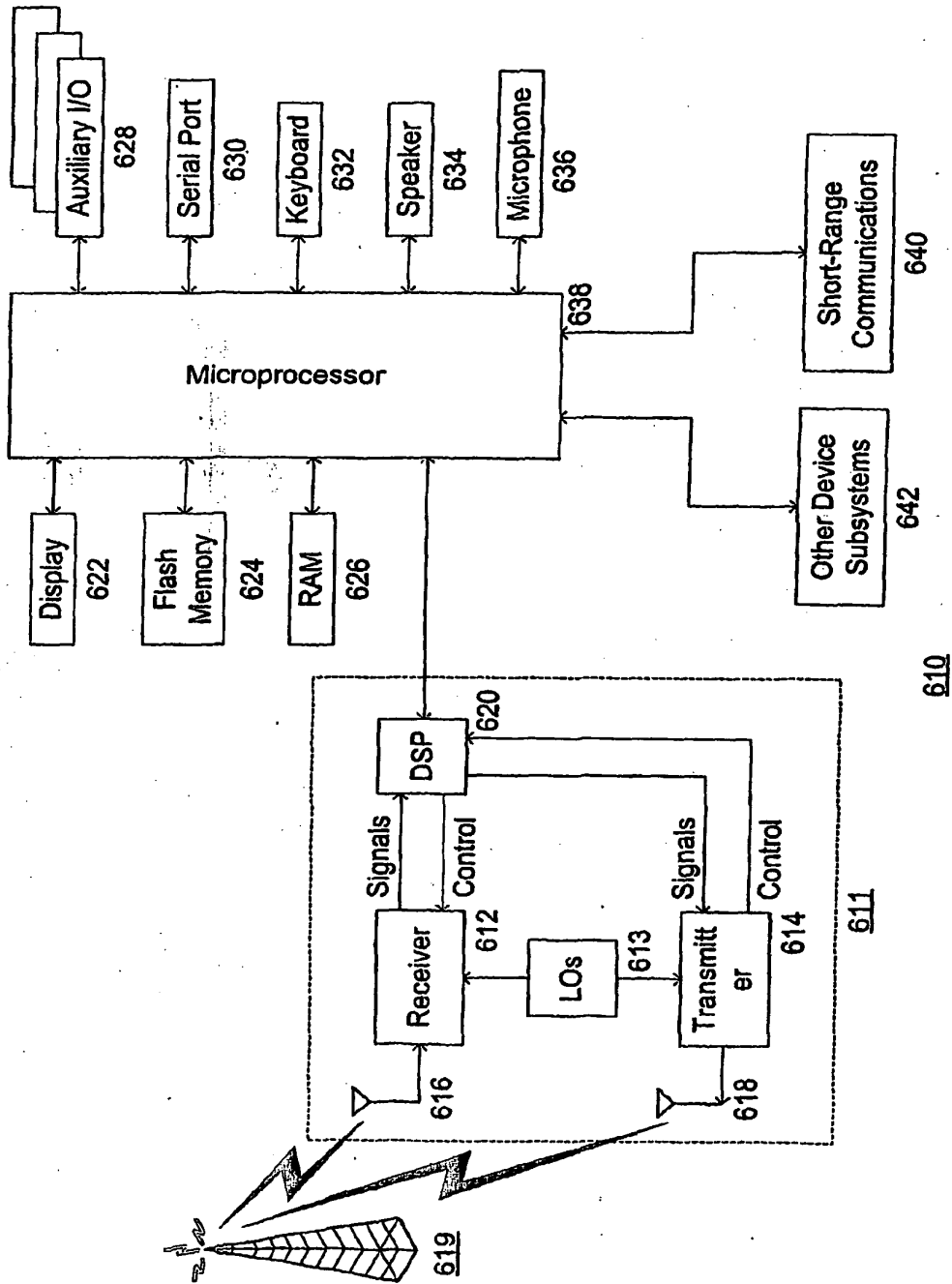


Figure 6

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 5978484 A [0003]



(11) **EP 2 306 259 A2**

(12) **EUROPEAN PATENT APPLICATION**

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

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

(21) Application number: **10186194.6**

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

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

- **Brown, Michael S.**
Heidelberg Ontario (CA)
- **Little, Herbert A**
Waterloo Ontario (CA)

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

(74) Representative: **Finnie, Peter John**
Gill Jennings & Every LLP
The Broadgate Tower
20 Primrose Street
London EC2A 2ES (GB)

(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, ON N2L 3W8 (CA)

Remarks:

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

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

(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 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.

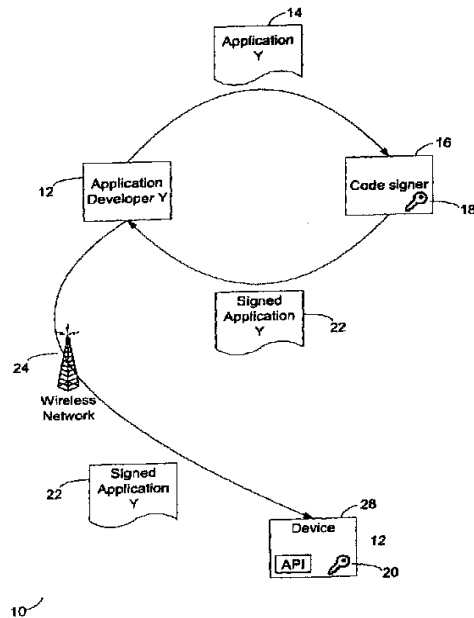


Figure 1

EP 2 306 259 A2

DescriptionCROSS-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.

BACKGROUND1. 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.

[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 I-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

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.

[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 may 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 aspects according to the present invention.

Numbered Clause 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).

Numbered Clause 2. The code signing system of Numbered Clause 1, wherein

- (i) the virtual machine (64) denies the software application (66) access to the API if the digital signature (96) is not authenticated, or
- (ii) wherein the virtual machine (64) purges the software application (66) if the digital signature (96) is not authenticated,

Numbered Clause 3. The code signing system of Numbered Clauses 1 or 2, wherein

- (iii) the code signing system is installed on a mobile device (62), or
- (iv) wherein the digital signature (96) is generated by a code signing authority.

Numbered Clause 4. The code signing system of

any of Numbered Clauses 1 to 3, further comprising:

- a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs, wherein the virtual machine (64) controls access to the plurality of API libraries by the software application (66).

Numbered Clause 5. The code signing system of any of Numbered Clauses 1 to 4,

- wherein a least one of the plurality of API libraries is classified as sensitive;
- wherein access to a sensitive API library requires 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) includes 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) authenticates 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.

Numbered Clause 6. The code signing system of Numbered Clauses 1 to 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).

Numbered Clause 7. The code signing system of Numbered Clause 6, 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.

Numbered Clause 8. The code signing system of Numbered Clause 3, wherein the API further comprises:

- a description string (88) that is displayed by the mobile device (62) when the software applica-

tion (66) attempts to access the API.

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

- (i) comprises an operating system (84), or
- (ii) comprises one or more core functions (86) of a mobile device (62), or
- (iii) comprises hardware (82) on a mobile device (62).

Numbered Clause 10. The code signing system of Numbered Clause 9, wherein the hardware (82) comprises a subscriber identity module (SIM) card.

Numbered Clause 11. The code signing system of any of Numbered Clauses 1 to 10, wherein the software application (66) is a Java application for a mobile device (62).

Numbered Clause 12. The code signing system of any of Numbered Clauses 1 to 11, wherein

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

Numbered Clause 13. The code signing system of any of Numbered Clauses 1 to 12, wherein the virtual machine (64) is a Java virtual machine installed on a mobile device (62).

Numbered Clause 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).

Numbered Clause 15. The method of Numbered Clause 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).

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

Numbered Clause 17. The method of any of Numbered Clauses 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 sensitive API where the digital signature (96) is not authenticated.

Numbered Clause 18. The method of Numbered Clause 17, comprising the additional step of:

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

Numbered Clause 19. The method of Numbered Clause 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.

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

Numbered Clause 21. The method of any of Numbered Clauses 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.

Numbered Clause 22. The method of Numbered Clause 21, comprising the additional step of:

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

Numbered Clause 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).

Numbered Clause 24. The mobile device of Numbered Clause 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.

Numbered Clause 25. The mobile device of Numbered Clauses 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.

Numbered Clause 26. The mobile device of Numbered Clauses 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).

Numbered Clause 27. The mobile device of Numbered Clauses 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.

Numbered Clause 28. The mobile device of Numbered Clause 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.

Numbered Clause 29. The mobile device of Numbered Clauses 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 method for restricting execution of a software application on a mobile device, the method comprising:

determining whether the software application is signed; and
requiring verification of at least a digital signature before the software application can be executed on the mobile device.
2. The method of claim 1, further comprising:

preventing execution of the software application if the digital signature cannot be verified.
3. The method of any of one of the preceding claims, further comprising:

purging the software application from the mobile device if the digital signature cannot be verified.
4. The method of any of one of the preceding claims, further comprising:

executing the software application on the mobile device if the digital signature is verified.
5. The method of any of one of the preceding claims, wherein the digital signature comprises a global signature.
6. The method of claim 1, wherein the digital signature corresponds to a specific resource of the mobile de-

vice.

7. The method of claim 6, further comprising:

denying the software application access to the specific resource when the software application does not have a digital signature corresponding to the specific resource.
8. The method of claim 6, further comprising:

granting the software application access to the specific resource when the software application has a digital signature corresponding to the specific resource.
9. The method of any of one of the preceding claims, wherein the software application comprises an updated software application.
10. The method of any of one of the preceding claims, further comprising:

displaying a message stating that the software application is attempting to access one or more resources of the mobile device.
11. The method of any of one of the preceding claims, further comprising:

providing a user option for final control to either grant or deny access by the software application to one or more resources of the mobile device.
12. The method of any of one of the preceding claims, further comprising:

prompting for user authorization to execute the software application.
13. The method of any of one of the preceding claims, wherein the software application includes multiple signatures.
14. The method of claim 13, wherein the multiple signatures comprise at least one of the following: a global signature, a device-specific signature, a library-specific signature, or an API-specific signature or a combination thereof.
15. The method of claim 13, wherein access by the software application to one or more APIs or libraries of the mobile device is restricted dependent upon verification of respective signatures corresponding to the one or more APIs or libraries.
16. The method of claim 15, wherein the one or more APIs or libraries comprise at least one of the follow-

ing: a database API, a user interface API, a cryptographic API, a radio API, an API that interfaces with a calendar program and calendar entry data models, an API that interfaces with the transmission circuitry and functions of the mobile device, an API capable of interfacing with lower-level services performed by the mobile device's operating system, or a combination thereof. 5

17. The method of any one of the preceding claims, wherein the software applications comprises a virus or malicious code. 10

18. A device for performing the method of any one of claims 1 through 17. 15

19. Code for performing the method of any one of claims 1 through 17.

20

25

30

35

40

45

50

55

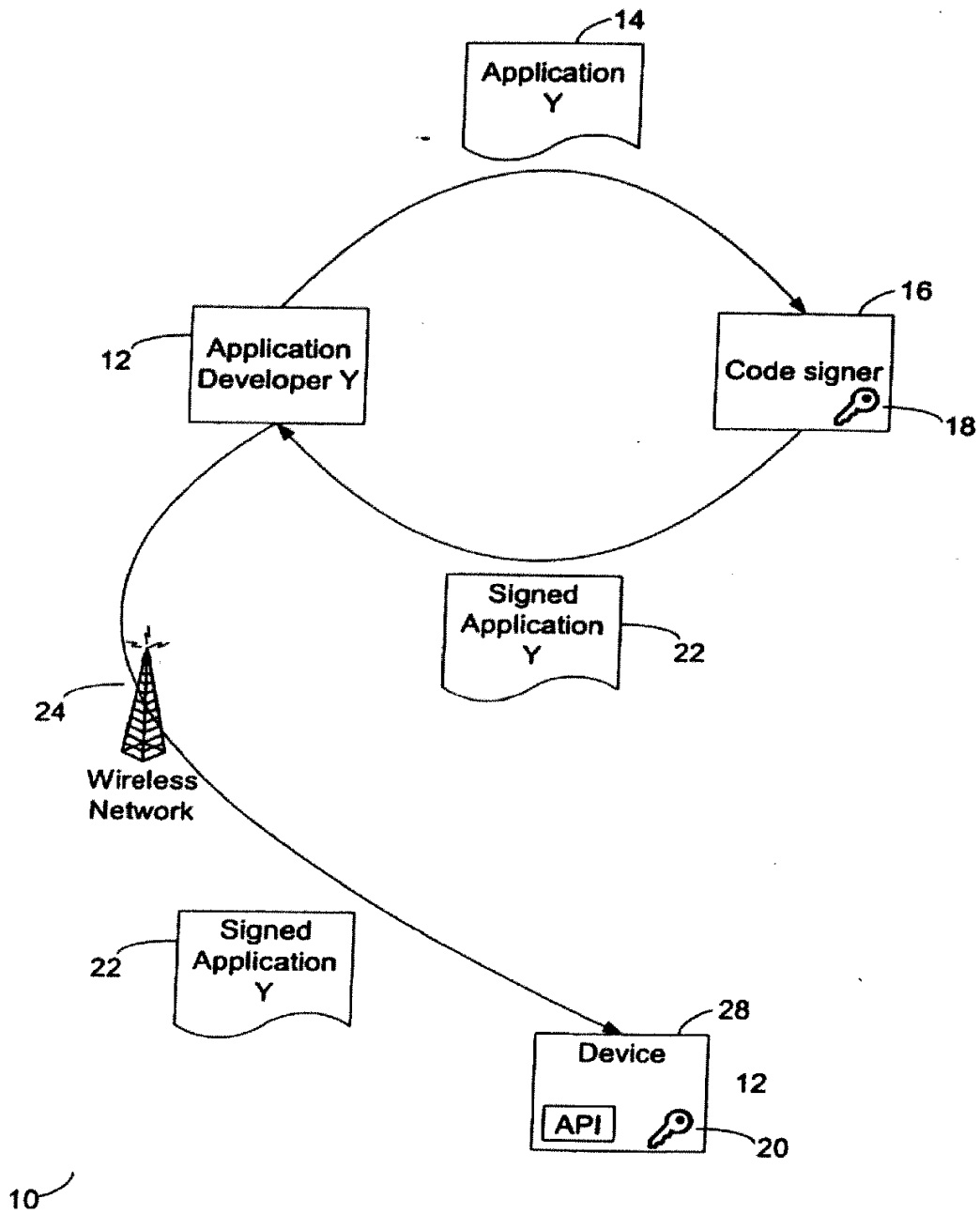
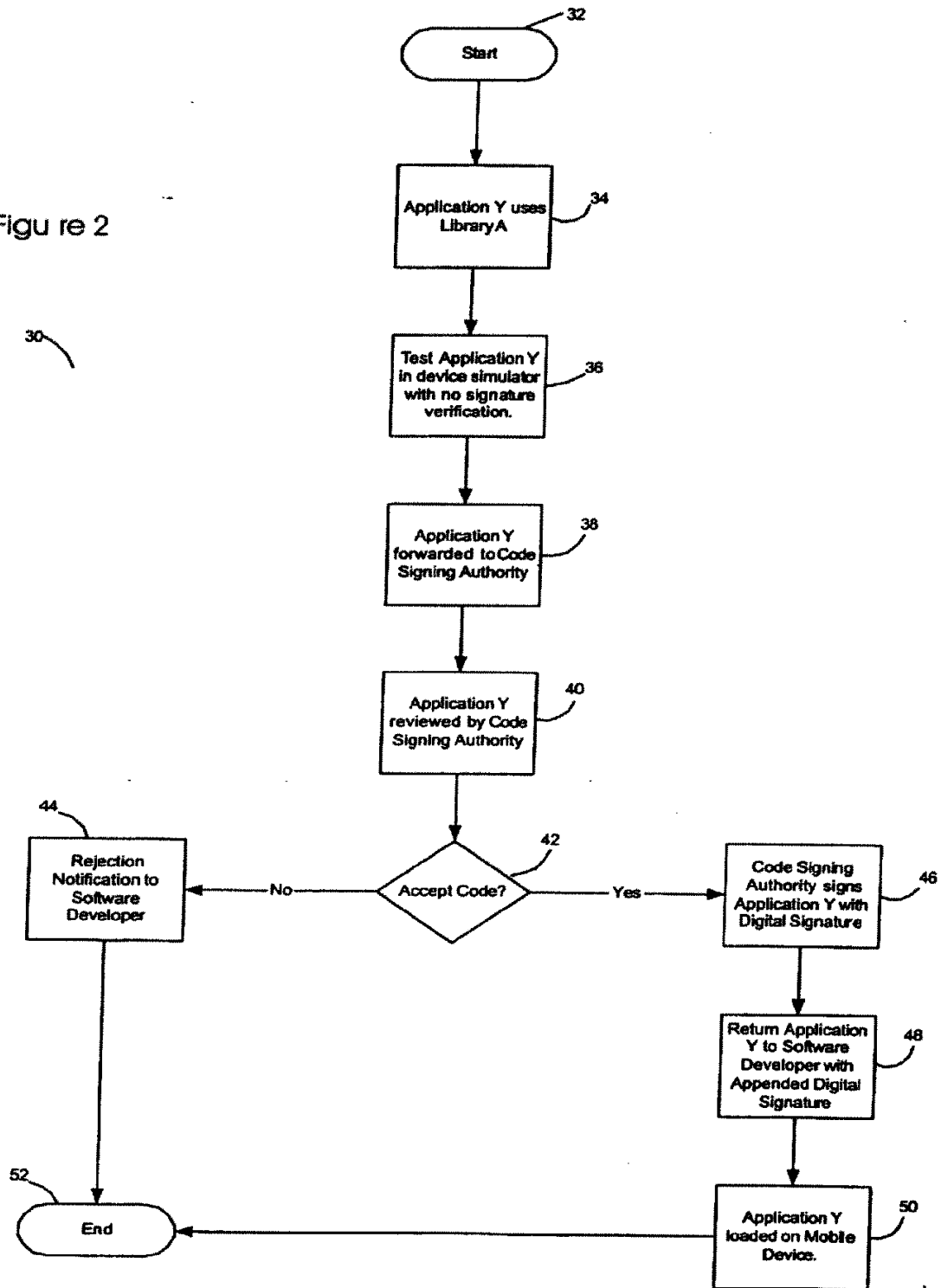
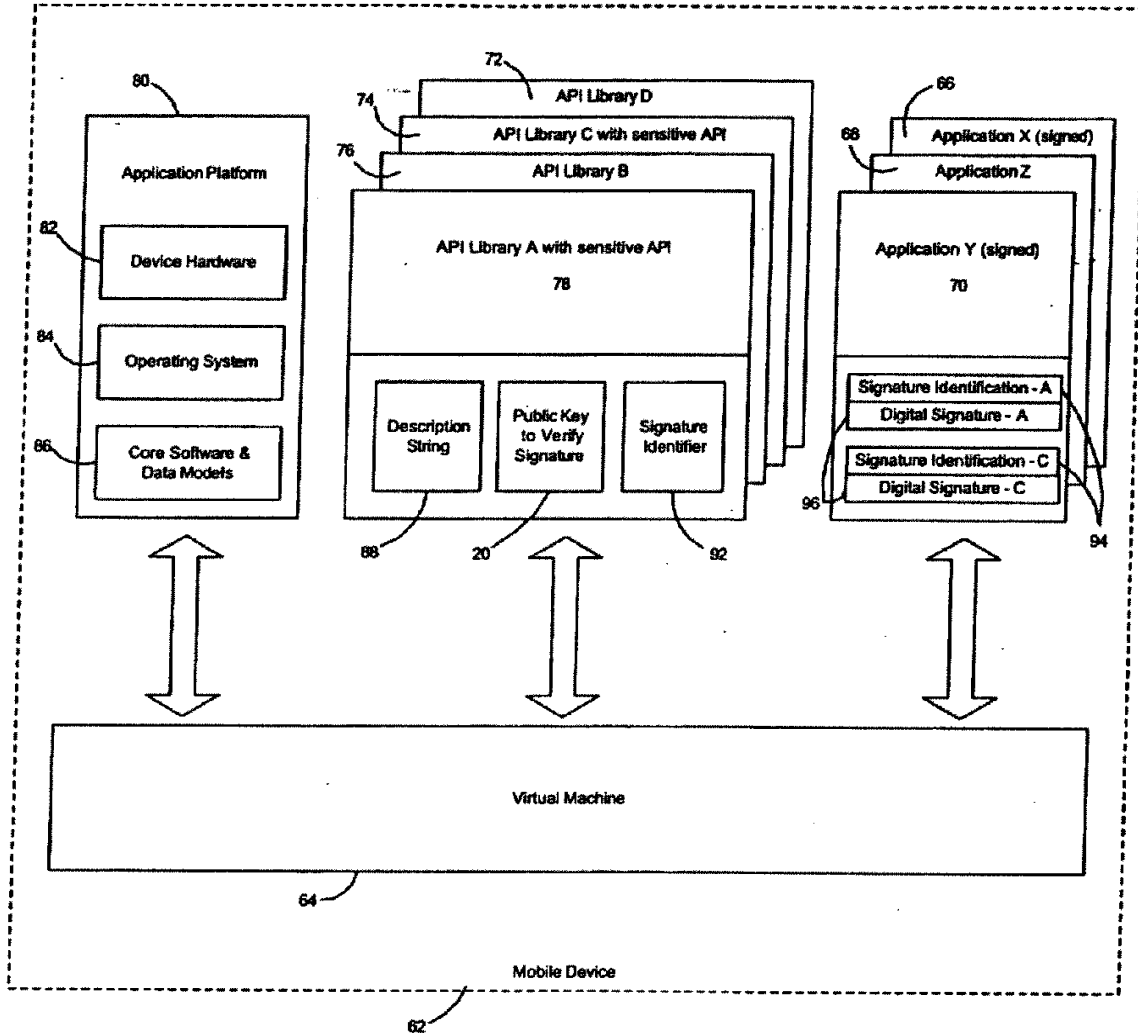


Figure 1

Figure 2





60

Figure 3

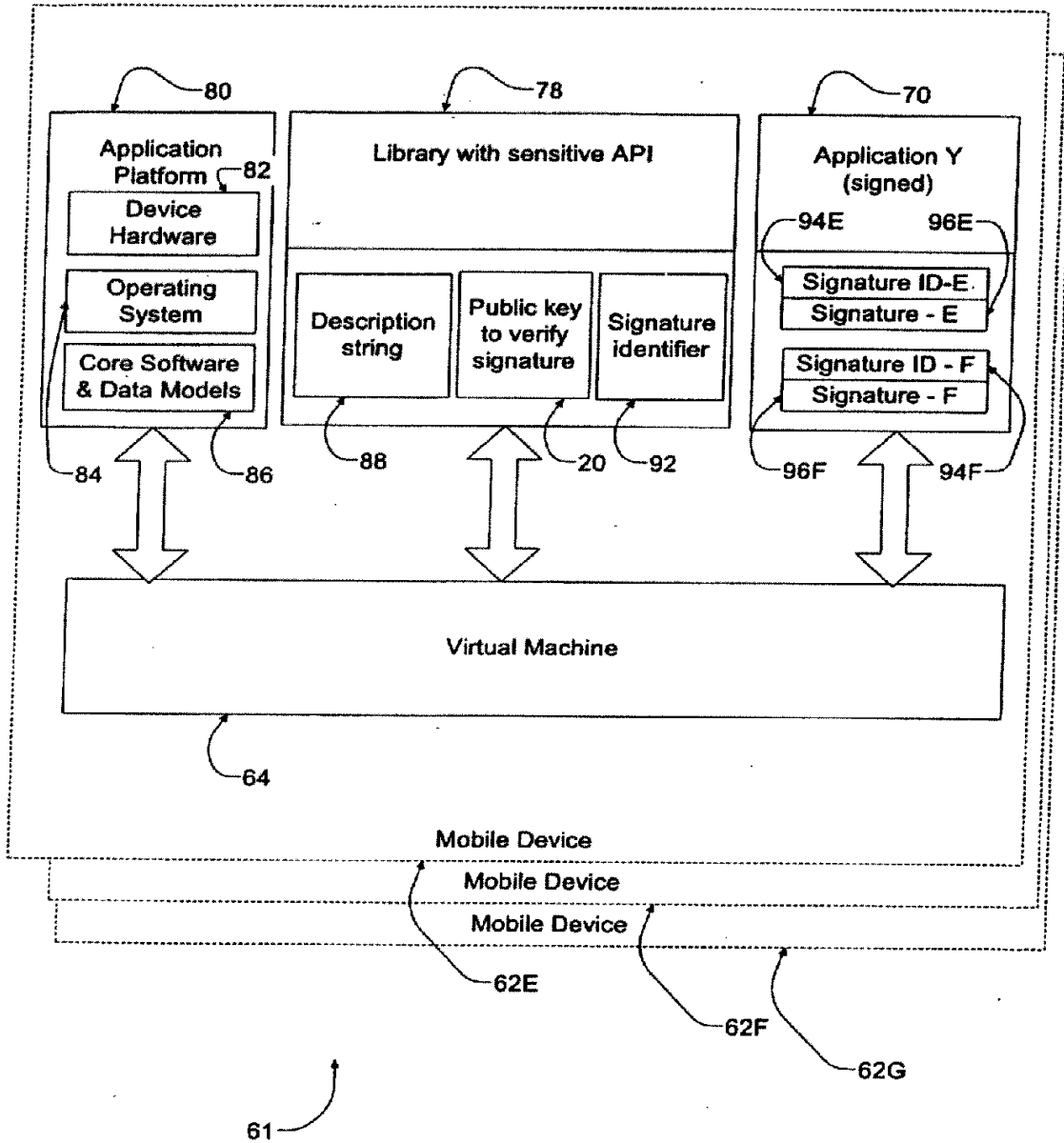
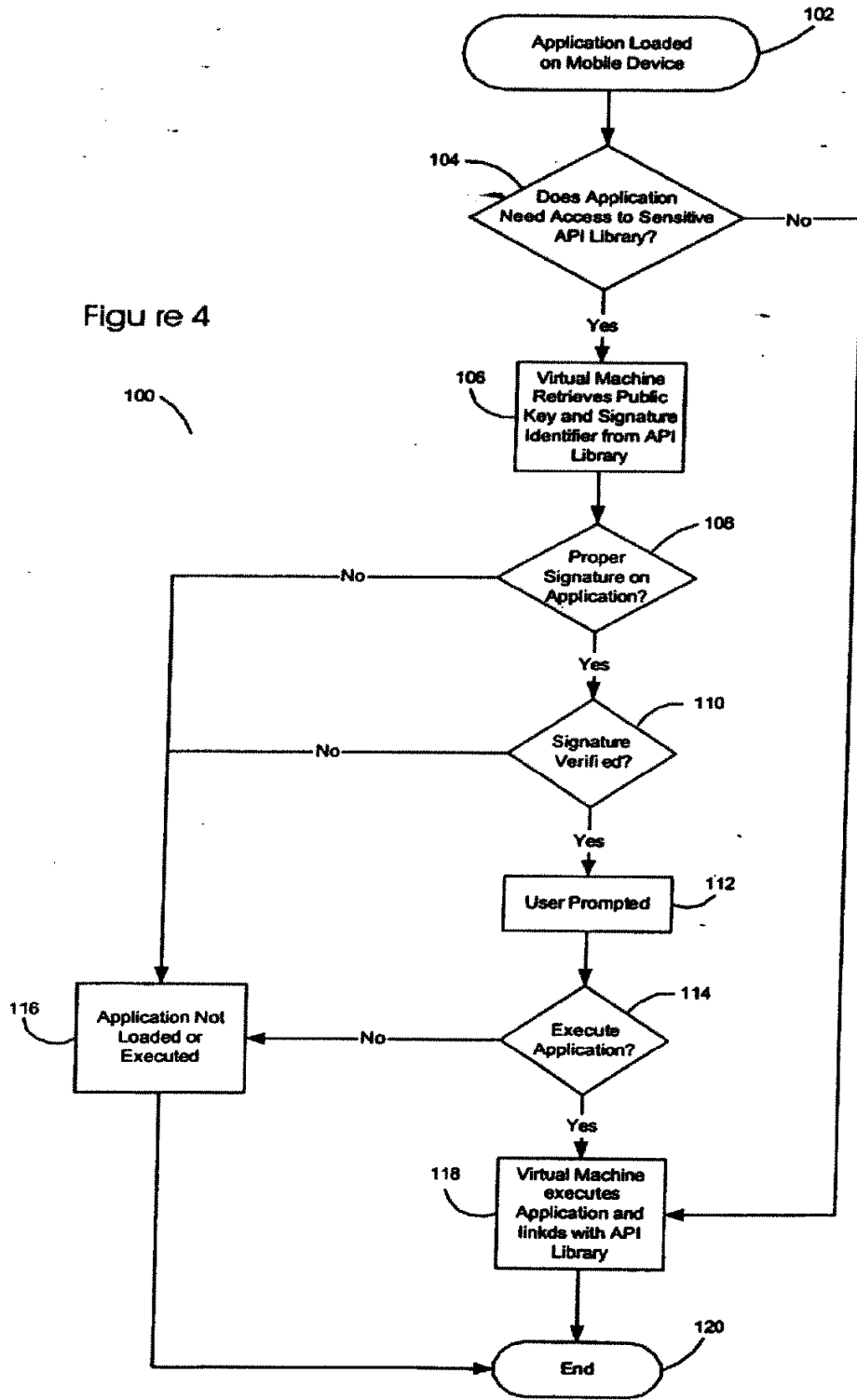


Figure 3A

Figure 4



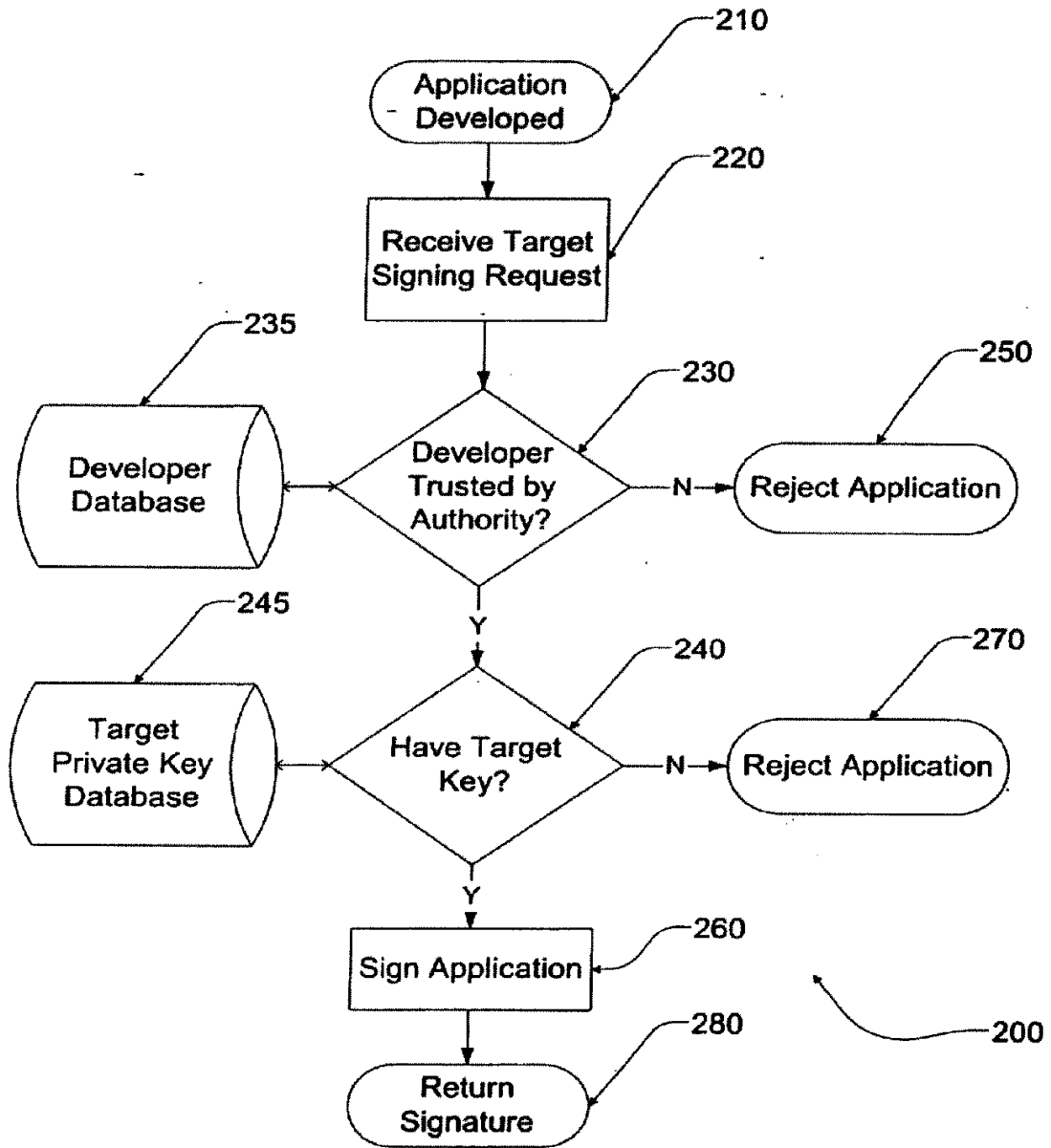


Figure 5

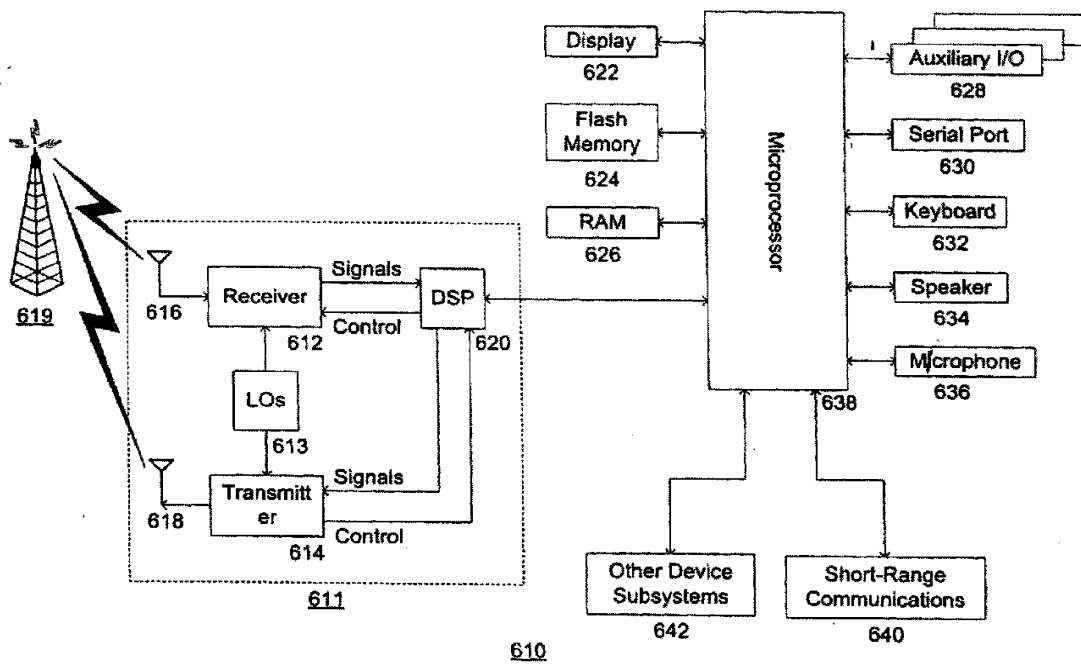


Figure 6

EP 2 306 259 A2

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]

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication: **15.02.2006 Bulletin 2006/07** (51) Int Cl.: **G06F 1/00 (2006.01)**

(21) Application number: **05024661.0**

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

(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

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
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)

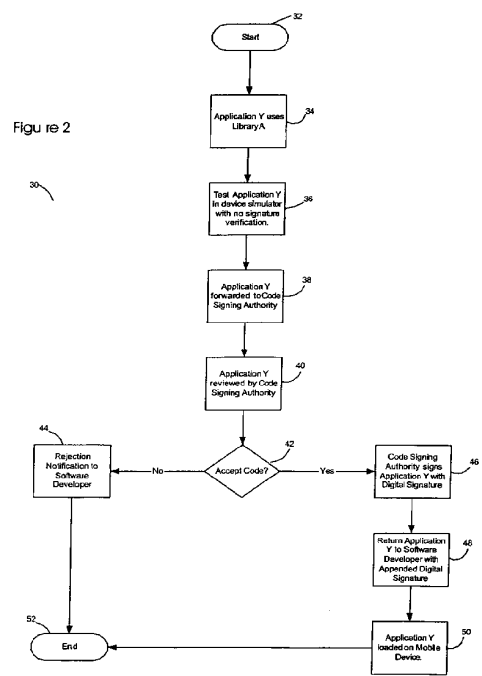
• **Little, Herbert A.**
Waterloo,
Ontario N2T 2V8 (CA)

(74) Representative: **Jones Day**
Rechtsanwälte, Attorneys-at-Law, Patentanwälte
Prinzregentenstrasse 11
80538 München (DE)

Remarks:
 This application was filed on 11 - 11 - 2005 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 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.



EP 1 626 324 A2

DescriptionCROSS-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.

BACKGROUND1. 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 2.5 G 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.

[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 in-

puts. 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

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 628, 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 may 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.

Claims

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 claim 1, wherein
 - (i) the virtual machine (64) denies the software application (66) access to the API if the digital signature (96) is not authenticated, or
 - (ii) wherein the virtual machine (64) purges the software application (66) if the digital signature (96) is not authenticated,
3. The code signing system of claim 1 or 2, wherein
 - (iii) the code signing system is installed on a mobile device (62), or
 - (iv) wherein the digital signature (96) is generated by a code signing authority.
4. The code signing system of any of claims 1 to 3, further comprising:

- a plurality of API libraries, each of the plurality of API libraries includes a plurality of APIs, wherein the virtual machine (64) controls access to the plurality of API libraries by the software application (66). 5
5. The code signing system of any of claims 1 to 4,
- wherein at least one of the plurality of API libraries is classified as sensitive; 10
 - wherein access to a sensitive API library requires 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; 15
 - wherein the software application (66) includes at least one digital signature (96) and at least one associated signature identification (94) for accessing sensitive API libraries; and 20
 - wherein the virtual machine (64) authenticates 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. 25
6. The code signing system of any of claims 1 to 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). 30
7. The code signing system of claim 6, wherein: 35
- the digital signature (96) is generated by applying the private signature key to a hash of the software application (66); and 40
 - 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. 45
8. The code signing system of claim 3, wherein the API further comprises: 50
- 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 claims 1 to 7, wherein the application platform 55
- (i) comprises an operating system (84), or
 - (ii) comprises one or more core functions (86) of a mobile device (62), or
 - (iii) comprises hardware (82) on a mobile device (62).
10. The code signing system of claim 9, wherein the hardware (82) comprises a subscriber identity module (SIM) card.
11. The code signing system of any of claims 1 to 10, wherein the software application (66) is a Java application for a mobile device (62).
12. The code signing system of any of claims 1 to 11, wherein
- (i) the API interfaces with a cryptographic routine on the application platform, or wherein
 - (ii) the API interfaces with a proprietary data model on the application platform. 20
13. The code signing system of any of claims 1 to 12, wherein the virtual machine (64) is a Java virtual machine installed on a mobile device (62). 25
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). 35
15. The method of claim 14, comprising the additional step of: 45
- 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 claim 14 or claim 15, wherein the digital signature (96) and the signature identification (94) are generated by a code signing authority. 50
17. The method of any of claims 14 to 16, comprising the additional steps of: 55
- 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 claim 17, 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 claim 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.
20. The method of claim 19, wherein the digital signature (96) is generated by calculating a hash of the software application (66) and applying the private signature key.
21. The method of any of claims 14 to 20, 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 claim 21, 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 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).
24. The mobile device of claim 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.
25. The mobile device of claim 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.
26. The mobile device of any of claims 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).
27. The mobile device of any of claims 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.
28. The mobile device of claim 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 (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 claims 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) at-
tempts to access said at least one of the APIs.

5

10

15

20

25

30

35

40

45

50

55

15

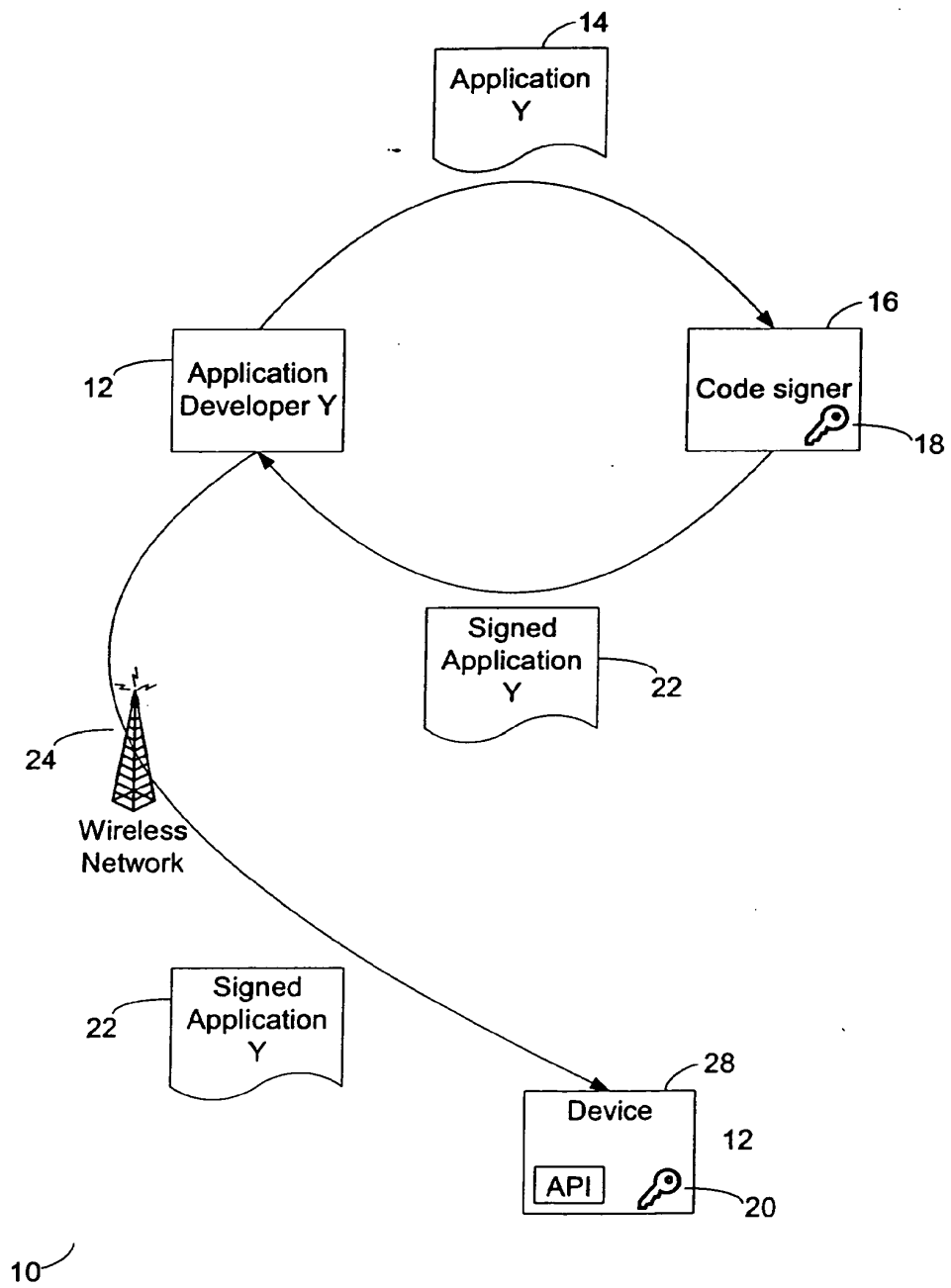
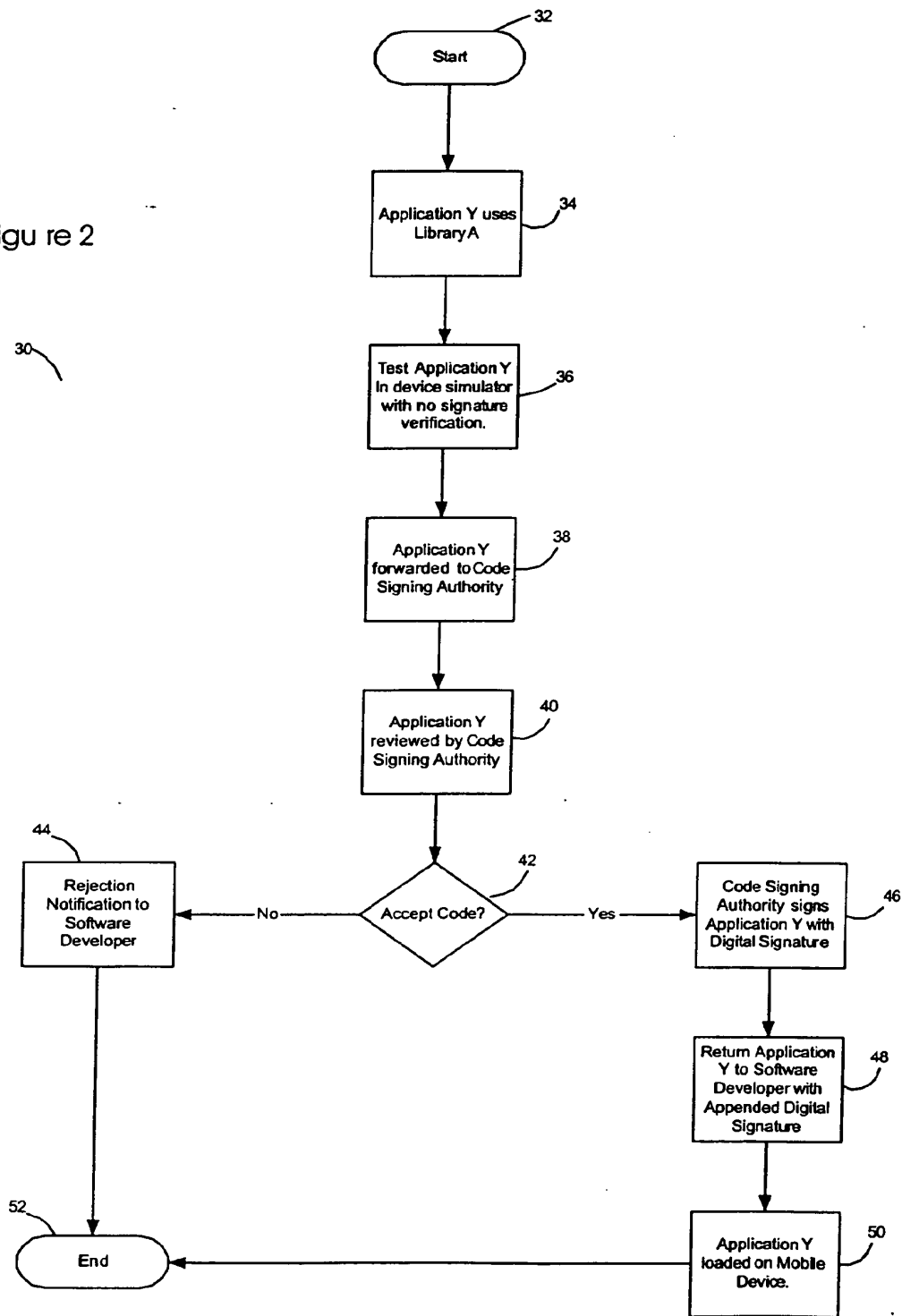
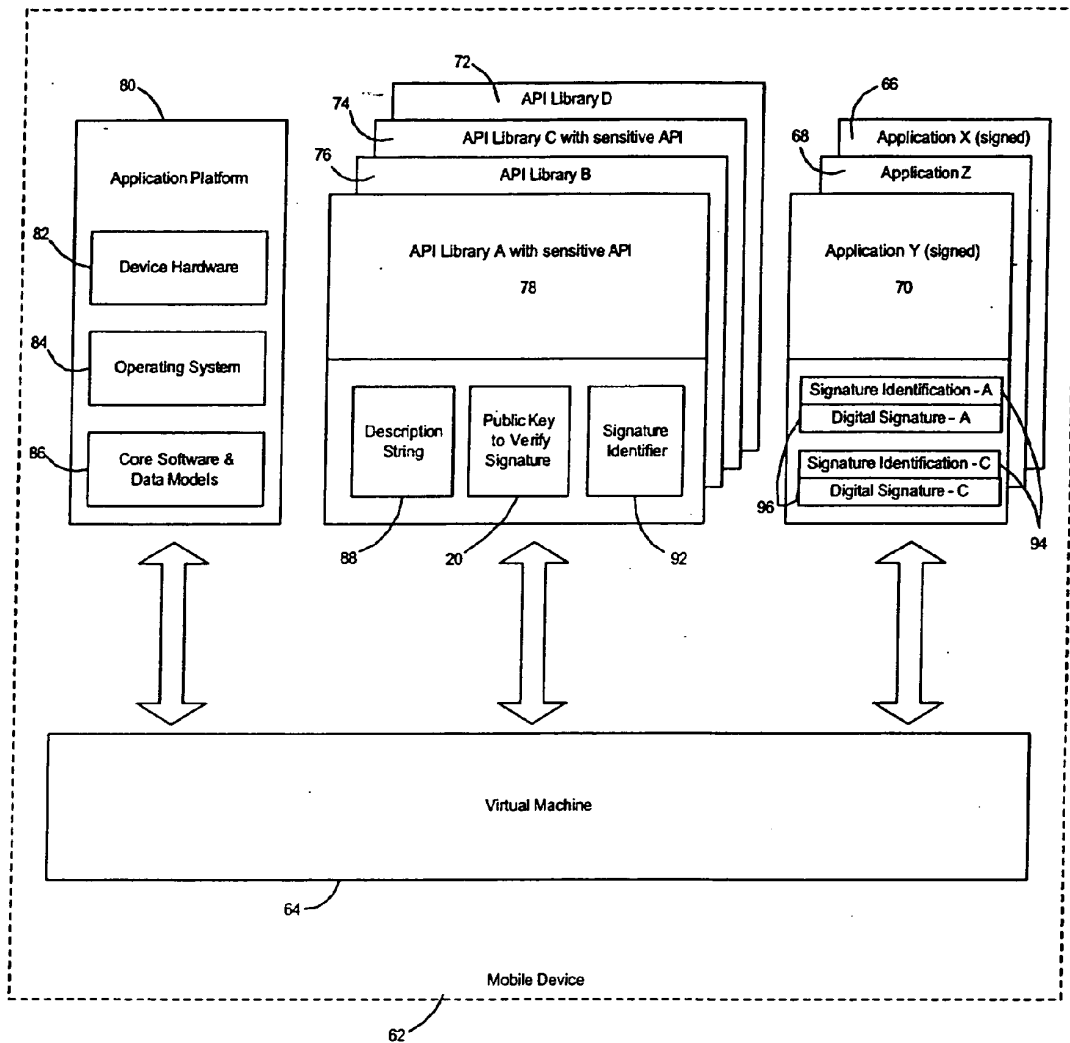


Figure 1

Figure 2





60

Figure 3

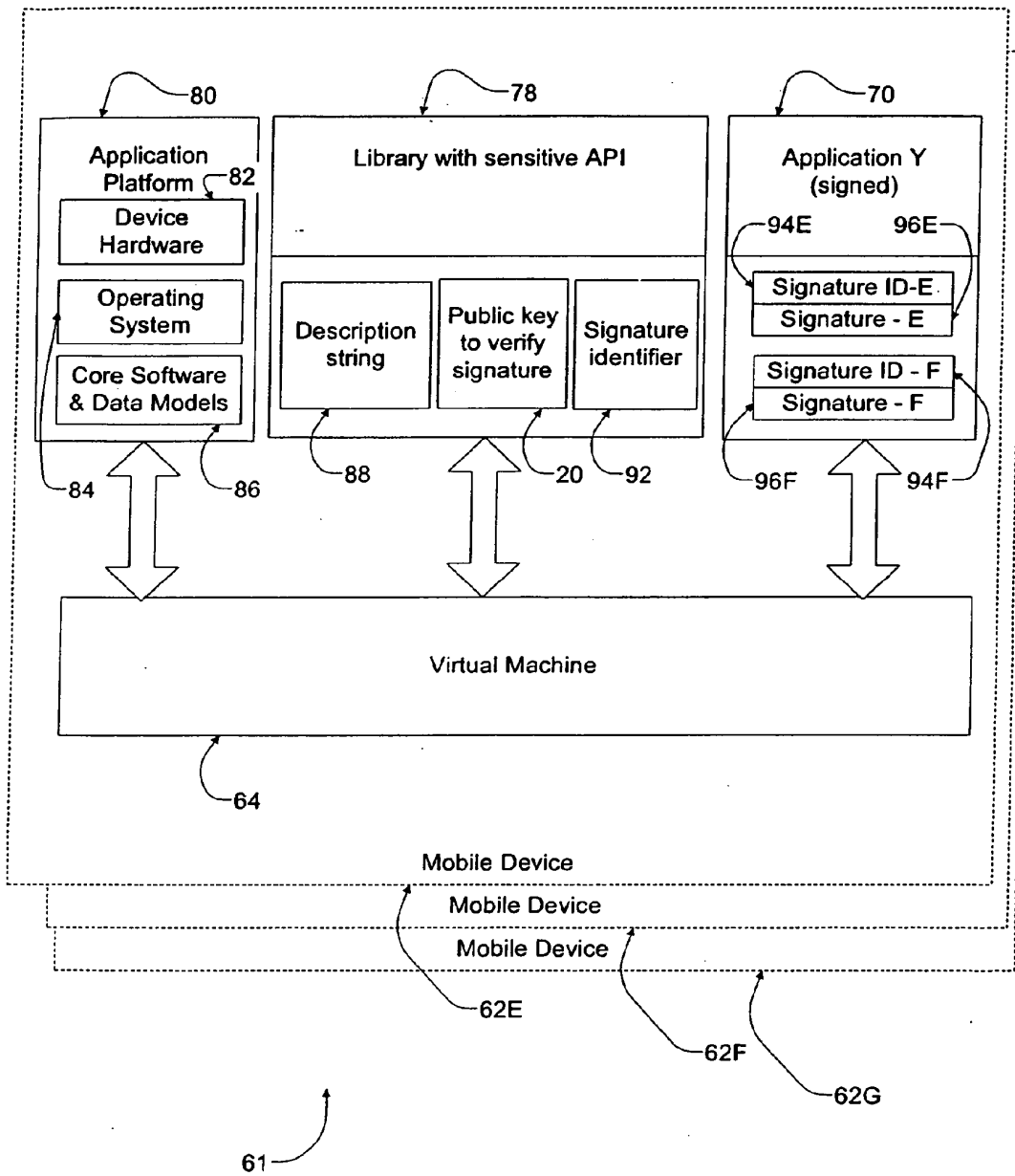
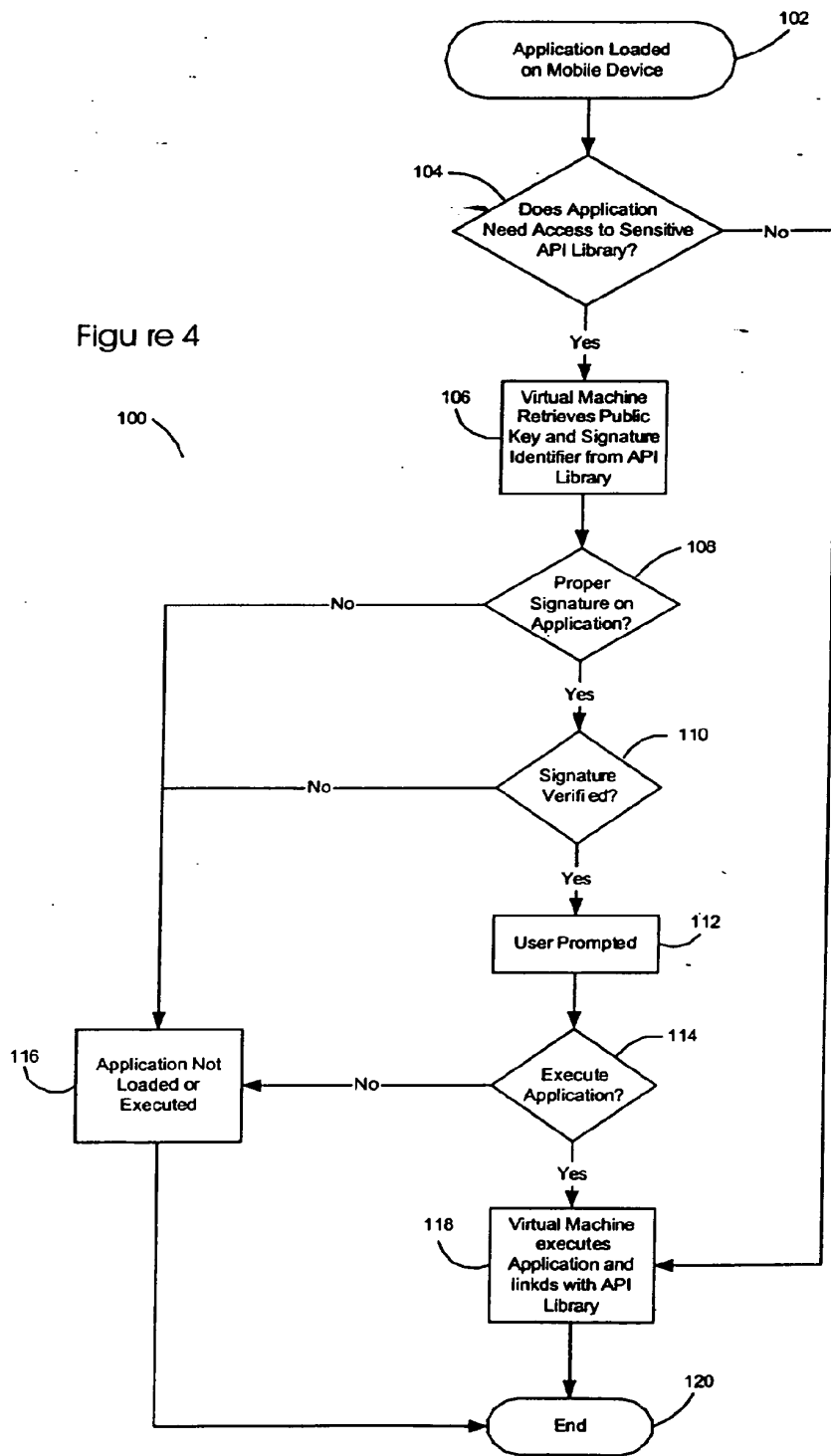


Figure 3A

Figure 4



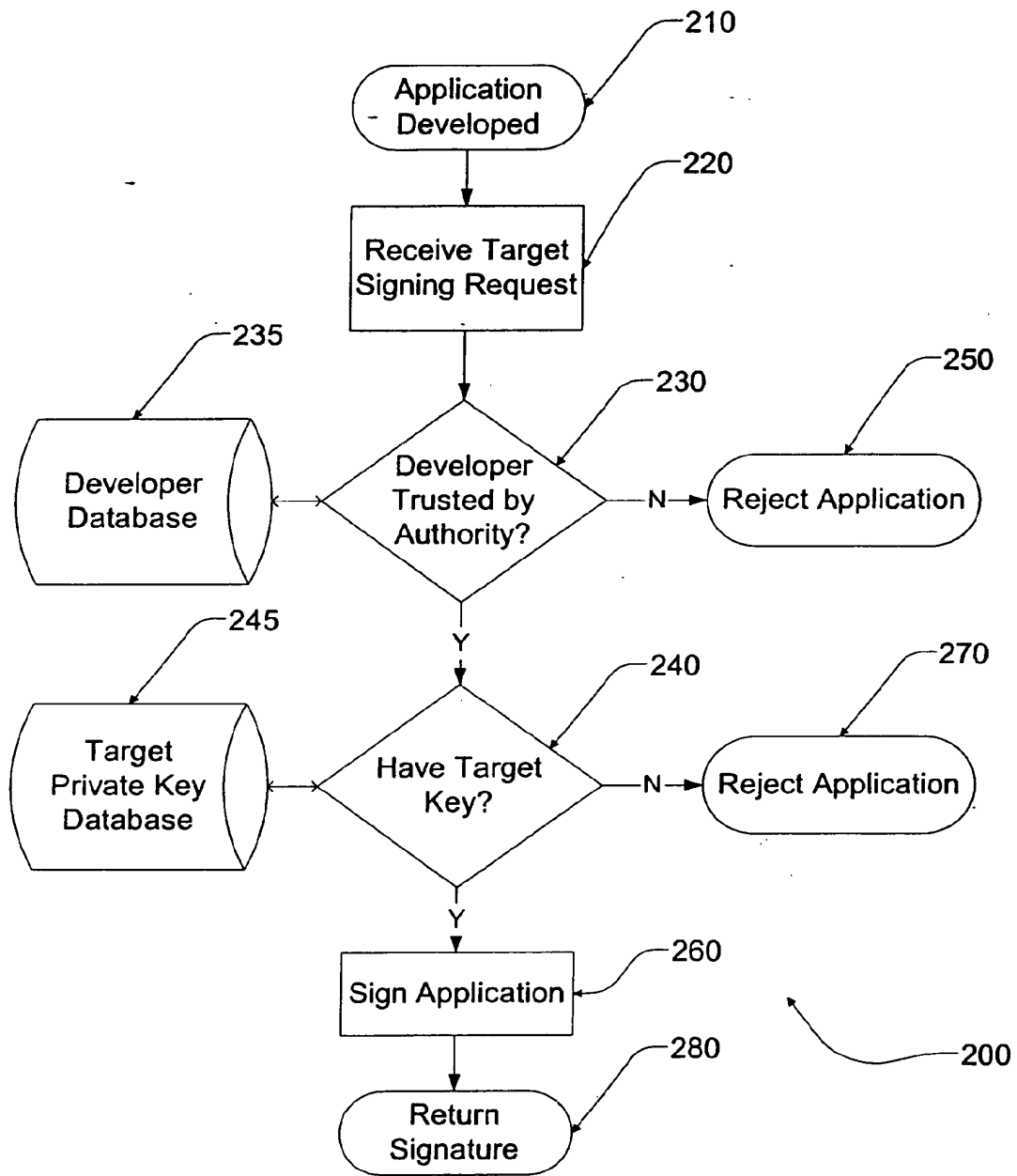


Figure 5

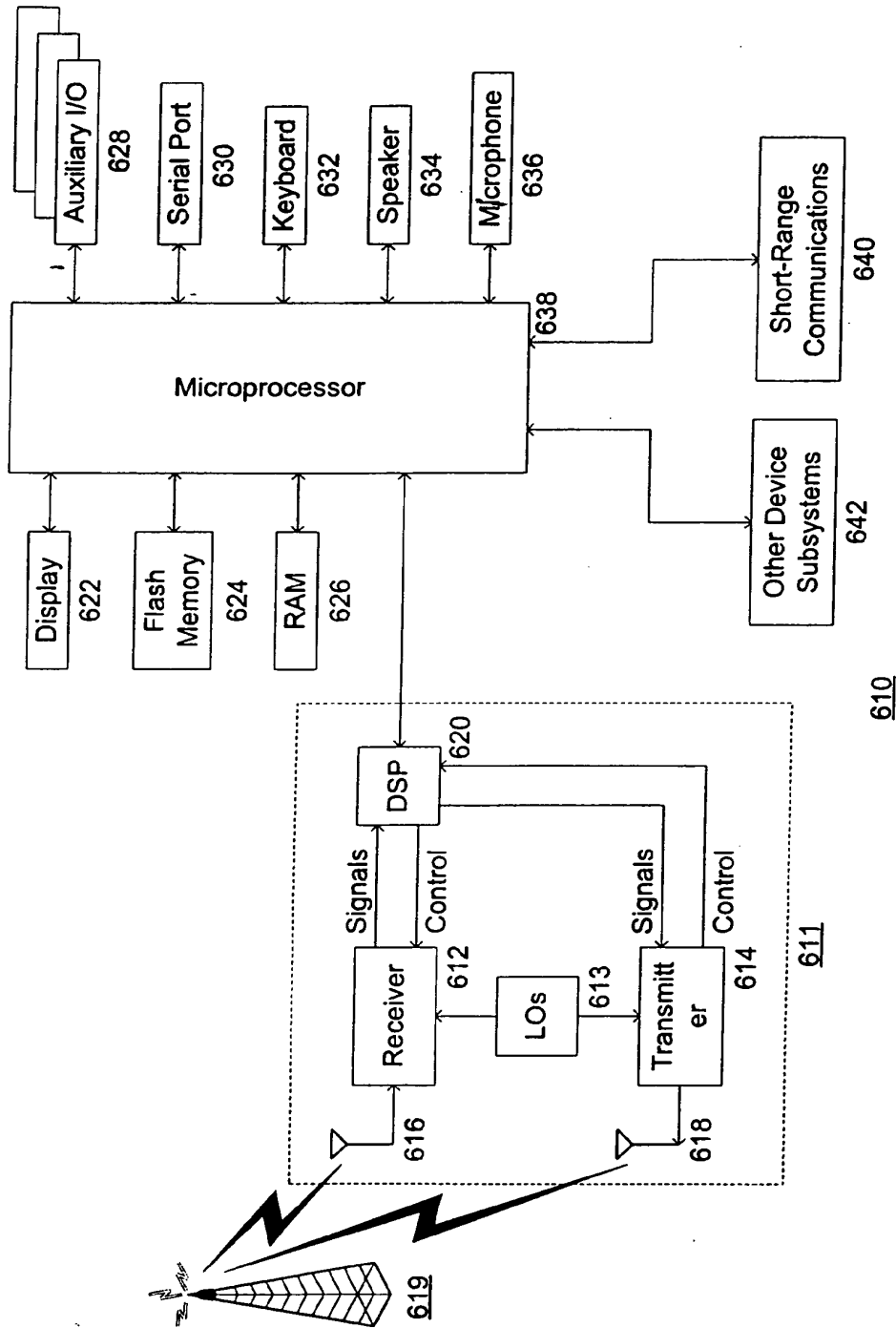


Figure 6



(11) **EP 2 284 644 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
16.02.2011 Bulletin 2011/07

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

(21) Application number: **10183655.9**

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

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

- **Brown, Michael S
Heidelberg (CA)**
- **Little, Herbert A
Waterloo Ontario (CA)**

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

(74) Representative: **Finnie, Peter John
Gill Jennings & Every LLP
The Broadgate Tower
20 Primrose Street
London EC2A 2ES (GB)**

(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)**

Remarks:

This application was filed on 30-09-2010 as a divisional application to the application mentioned under INID code 62.

(72) Inventors:
• **Yach, David P
Waterloo Ontario (CA)**

(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 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.

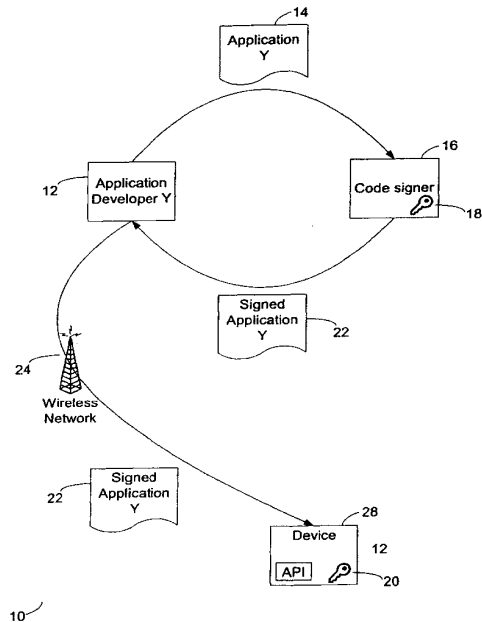


Figure 1

EP 2 284 644 A1

DescriptionCROSS-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.

BACKGROUND1. 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 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 in-

puts. 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

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.

[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 may 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

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.