US006766353B1

(12) **United States Patent**
Lin et al.

(10) **Patent No.: US 6,766,353 B1**
(45) **Date of Patent: Jul. 20, 2004**

(54) **METHOD FOR AUTHENTICATING A JAVA ARCHIVE (JAR) FOR PORTABLE DEVICES**

(75) Inventors: **Jyh-Han Lin**, Coral Springs, FL (US); **Robert L. Geiger**, Sunnyvale, CA (US); **Ronald R. Smith**, Coral Springs, FL (US); **Alan W. Chan**, Sunrise, FL (US); **Sanjay Wanchoo**, Lauderdhill, FL (US)

(73) Assignee: **Motorola, Inc.**, Schaumburg, IL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 411 days.

(21) Appl. No.: **09/613,804**

(22) Filed: **Jul. 11, 2000**

(51) **Int. Cl.**^{7} **G06F 15/16**

(52) **U.S. Cl.** **709/203**; 709/201

(58) **Field of Search** 713/200, 201, 713/156, 168; 709/200, 203, 225, 229, 226, 217

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,987,608 A		11/1999	Roskind	....	713/200
6,023,764 A		2/2000	Curtis	....	713/200
6,029,000 A		2/2000	Woolsey et al.	....	395/705
6,044,467 A		3/2000	Gong	....	713/200
6,341,353 B1	*	1/2002	Herman et al.	....	713/201
6,351,816 B1	*	2/2002	Mueller et al.	....	713/201
6,378,075 B1	*	4/2002	Goldstein et al.	....	713/200
6,381,696 B1	*	4/2002	Doyle	....	713/156
6,477,647 B1	*	11/2002	Venkatraman et al.	......	713/193
6,523,067 B2	*	2/2003	Mi et al.	....	709/229
6,606,708 B1	*	8/2003	Devine et al.	....	713/201

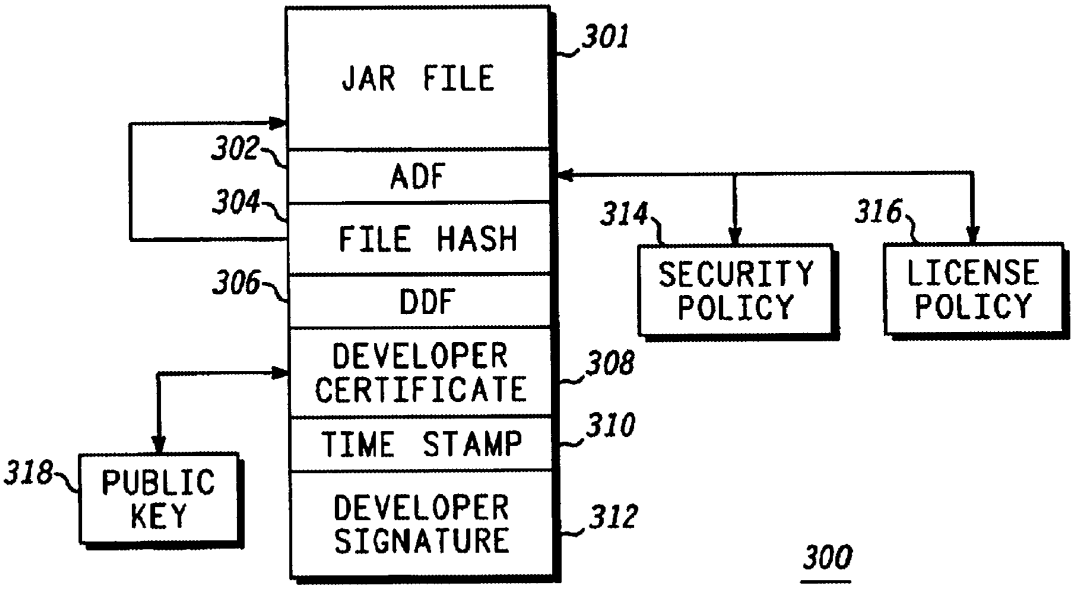
* cited by examiner

Primary Examiner—Mehmet B. Geckil
(74) *Attorney, Agent, or Firm*—Scott M. Garrett

(57) **ABSTRACT**

A signed application descriptor file **(206)** is used instead of X.509 certificates to authenticate a portable application code, such as a JAVA archive (JAR) file. The signed ADF includes an application descriptor file **(302)**, file hash **(304)** of the JAR file **(301)**, a developer descriptor file **(308)**, signed time stamp **(310)**, and a developer's certificate **(312)**. A network client device **(202)** includes limited computing resources **(212)** and a virtual machine environment for executing the portable code **(208)**. Furthermore the client device contains a set of cryptographic, digital keys for authenticating parts of the signed ADF, which are further used to authenticate the JAR file.

8 Claims, 3 Drawing Sheets



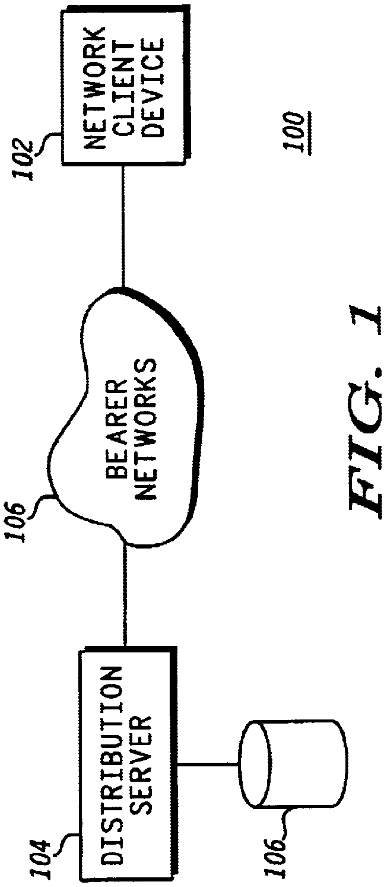

NETWORK CLIENT DEVICE
102
BEARER NETWORKS
106
DISTRIBUTION SERVER
104
106
100

FIG. 1

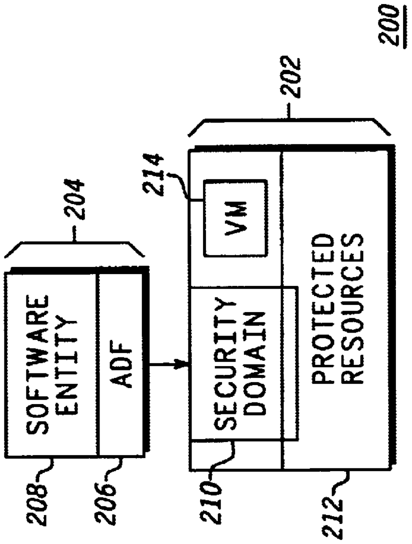

SOFTWARE ENTITY
ADF
204
208
206
SECURITY DOMAIN
VM
PROTECTED RESOURCES
214
202
210
212
200

FIG. 2

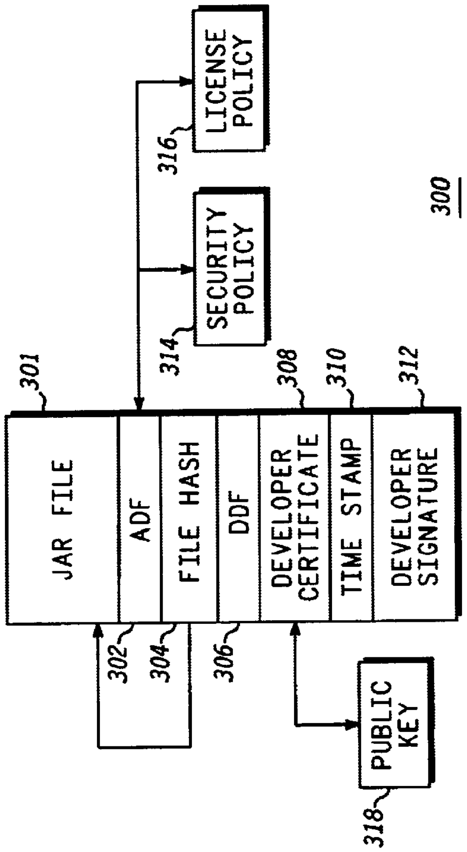

JAR FILE
ADF
FILE HASH
DDF
DEVELOPER CERTIFICATE
TIME STAMP
DEVELOPER SIGNATURE
SECURITY POLICY
LICENSE POLICY
PUBLIC KEY
300
301
302
304
306
308
310
312
314
316
318

FIG. 3

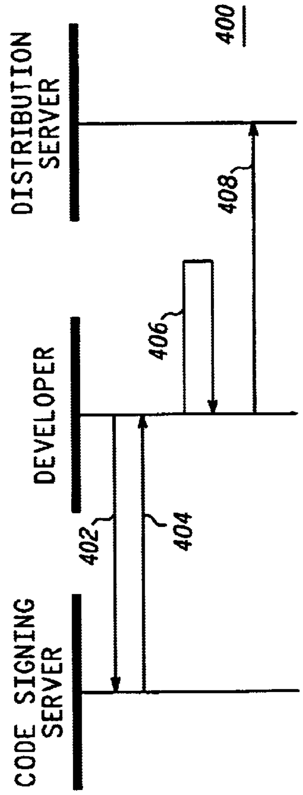

CODE SIGNING SERVER
DEVELOPER
DISTRIBUTION SERVER
400
402
404
406
408

FIG. 4

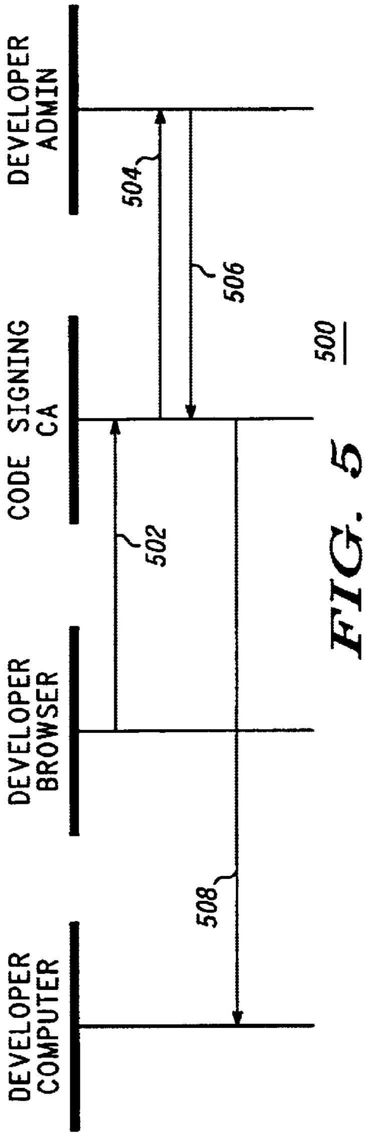

DEVELOPER COMPUTER
DEVELOPER BROWSER
CODE SIGNING CA
DEVELOPER ADMIN
502
504
506
508
500

FIG. 5

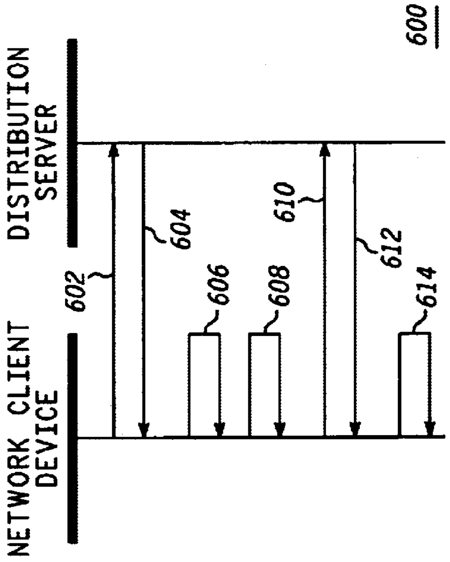

NETWORK CLIENT DEVICE
DISTRIBUTION SERVER
602
604
606
608
610
612
614
600

FIG. 6

METHOD FOR AUTHENTICATING A JAVA ARCHIVE (JAR) FOR PORTABLE DEVICES

TECHNICAL FIELD

This invention relates in general to portable code transfer, such as JAVA technology, and more particularly to security and authentication of portable code for use by wireless or mobile devices, or other computing devices with relatively limited computing resources, and limited communication bandwidth.

BACKGROUND OF THE INVENTION

In networked environments such as the Internet, the use of portable code or portable applications has gained widespread acceptance. The best known technology in this field is JAVA. In creating JAVA code, a developer creates an application and makes it available on a network in byte code format. The byte code is downloaded by various client devices connected to the network and loaded into a JAVA virtual machine environment on the client machine or computer. The virtual machine environment is a layer of software that can interact with the specific computing platform of the particular client device and interpret the byte code. An application so loaded onto a client device could compromise the client device, and may even be designed to do so if the developer of the application had malicious intentions. Therefore, security is a significant issue with portable code.

Many security schemes have been devised to address these security issues. These range from giving only very restricted access to all portable applications to a system of authentication in which different levels of permission may be granted depending on whether the application can be authenticated as having come from a trusted source. The later scheme is more preferable since it allows an application more access to the local computer's resources, so long as it is authenticated. This allows developers to create more powerful applications because the applications have more access to the computer resources of the client machine.

However, as presently devised, these authentication schemes are designed for general purpose personal computers, which are commonly referred to as "desktop" computers. These machines have varying degrees of computing resources, but in general the resources they have greatly exceed the computing resources of small, portable devices such as personal organizers and mobile communication devices. There is an increasing number of these smaller devices being manufactured that are able to connect to large networks, and particularly the internet. Presently X.509 certificates are widely used for authentication, but these are quite large files compared to the limited memory resources available on these smaller mobile devices. Furthermore, since the certificate comes bundled with the application typically, the device must load both the application and the certificate. Therefore a data structure and method of authenticating portable applications that can be used by smaller devices is needed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a typical network including a server and a client;

FIG. **2** shows a representation of a client network device and its computing resources;

FIG. **3** shows a block diagram of a signed application descriptor file (ADF) for use in accordance with the present invention;

FIG. **4** shows a sequence chart for creating a signed ADF in accordance with the invention;

FIG. **5** shows a sequence chart for establishing the identity of a trusted developer for use in creating a signed ADF in accordance with the invention; and

FIG. **6** shows a sequence chart for downloading a signed ADF and a portable application from a distribution server to a network client device in accordance with the invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

While the specification concludes with claims defining the features of the invention that are regarded as novel, it is believed that the invention will be better understood from a consideration of the following description in conjunction with the drawing figures, in which like reference numerals are carried forward. A brief description of the prior art is also thought to be useful.

The present invention solves the problem of downloading portable applications and authenticating their source onto client device with limited computing resources by creating a signed application descriptor file (ADF), and a developer descriptor file (DDF). The ADF is a file that describes the portable application in terms of the computing resources it requires, and can be loaded onto the client device first so that the client device can determine whether or not it has sufficient resources, or it can let the user of the client device determine if there are sufficient resources. The ADF file is signed by the developer of the corresponding application using a certification authority, which is a well known and trusted signing authority. Attributes in the signed ADF correspond to those of the application so that if the user of the client device decides to load the application, the application can be authenticated against the signed ADF. The DDF is associated with a particular application software developer and specifies the general access control related information assigned to the developer. For example, a DDF may restrict the kind of application libraries that applications developed by the developer can use, or the security domain to which the developer belongs.

Referring now to FIG. **1**, there is shown a typical network **100** over which client and server machines interact. In particular, a network client device **102**, such as a mobile communication device, connects with a distribution server **104** over one or more bearer networks **106**. Typically the bearer network includes a TCP/IP network, and for public distribution of software, it includes the Internet. However, numerous private networks are connected to the Internet through various gateways and portals, including many wireless mobile communication networks. Indeed, the present invention is suited particularly well to use on mobile communication devices such as Internet capable mobile or cellular radio telephones. These devices may use what is referred to as a "microbrowser" to view information, or "content", placed on the Internet and other networks accessible by the device, as well as execute portable code. As with general purpose computers, there is a desire to load portable applications onto these devices. Developers of portable applications provide the application on a database **106** of the distribution server **104**. Client devices access the distribution server over the network and receive the desired portable code or portable application over the connection. This is one way which JAVA code sections, such as applets, are distributed.

Referring to FIG. **2** there is shown a representation **200** of a client network device **202** and its computing resources. In

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.