US006009524A

United States Patent [19]

Olarig et al.

[11] **Patent Number: 6,009,524**

[45] **Date of Patent: Dec. 28, 1999**

[54] METHOD FOR THE SECURE REMOTE FLASHING OF A BIOS MEMORY

[75] Inventors: **Sompong P. Olarig**, Cypress; **Michael F. Angelo**, Houston, both of Tex.

[73] Assignee: **Compact Computer Corp**, Houston, Tex.

[21] Appl. No.: **08/920,810**

[22] Filed: **Aug. 29, 1997**

[51] **Int. Cl.**^{6} **G06F 9/24**

[52] **U.S. Cl.** **713/200**; 713/201; 709/225; 380/25

[58] **Field of Search** 709/225; 713/201, 713/200; 380/4, 25

[56] References Cited

U.S. PATENT DOCUMENTS

5,388,267	2/1995	Chan et al. .	
5,421,006	5/1995	Jablon et al. .	
5,455,865	10/1995	Perlman	380/49
5,666,416	9/1997	Micali	380/23
5,692,047	11/1997	McManis	380/4
5,757,914	5/1998	McManis	380/4
5,778,070	6/1996	Mattison	380/25
5,844,986	9/1996	Davis	380/4
5,859,911	6/1997	Angelo et al.	380/25

OTHER PUBLICATIONS

U.S.Patent Application "System and Method for Secure Information Transmission Over A Network", SN 08/764,177 filed Dec. 13, 1996, P–1257.

Primary Examiner—Ly V. Hua
Assistant Examiner—Wasseem Hamdan
Attorney, Agent, or Firm—Robert Groover

[57] ABSTRACT

An improved system and method for FLASH BIOS upgrades which is particularly useful in network hubs. Each hub or node which is equipped with a FLASH memory is also equipped with a validation system, which ensures that a received FLASH upgrade is authorized and uncorrupted. Each set of instructions to be flashed is marked both with a vendor authorization digital signature and also a system administrator authorization digital signature, and BOTH digital signatures must be recognized by the validation system before the FLASH memory will be upgraded. Because digital signatures are used for security purposes, flash upgrades can be performed from any location on the network, and are not limited to an administrative node.

17 Claims, 2 Drawing Sheets

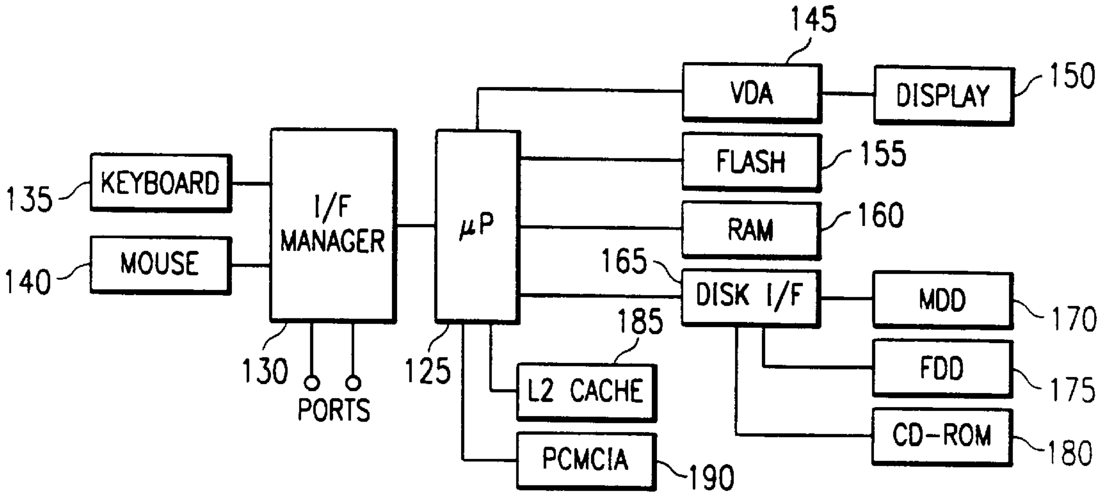


FIG. 1

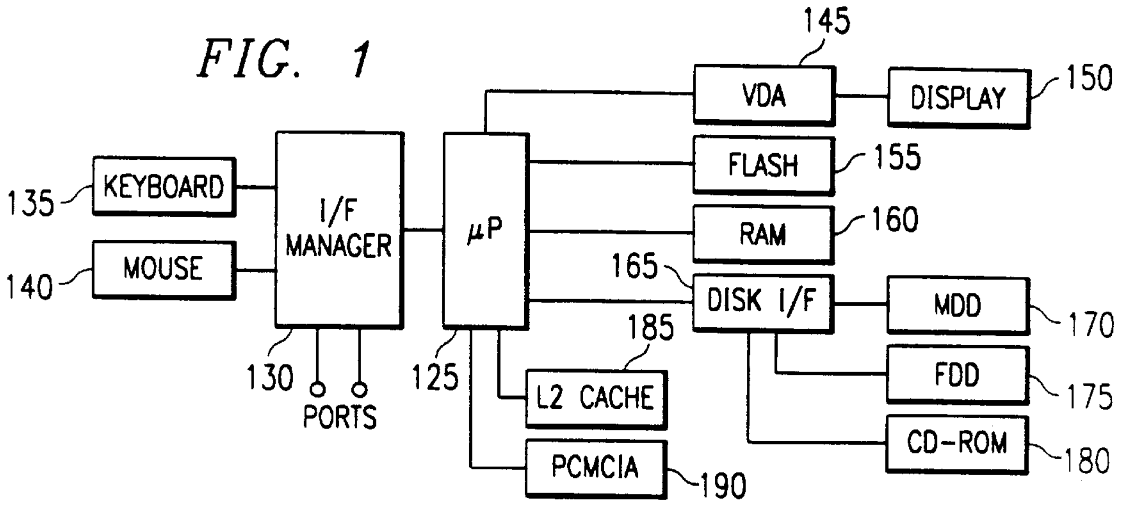

145
VDA
DISPLAY
150
FLASH
155
RAM
160
KEYBOARD
135
MOUSE
140
I/F MANAGER
μP
165
DISK I/F
MDD
170
185
FDD
175
130
PORTS
125
L2 CACHE
CD-ROM
180
PCMCIA
190

FIG. 2

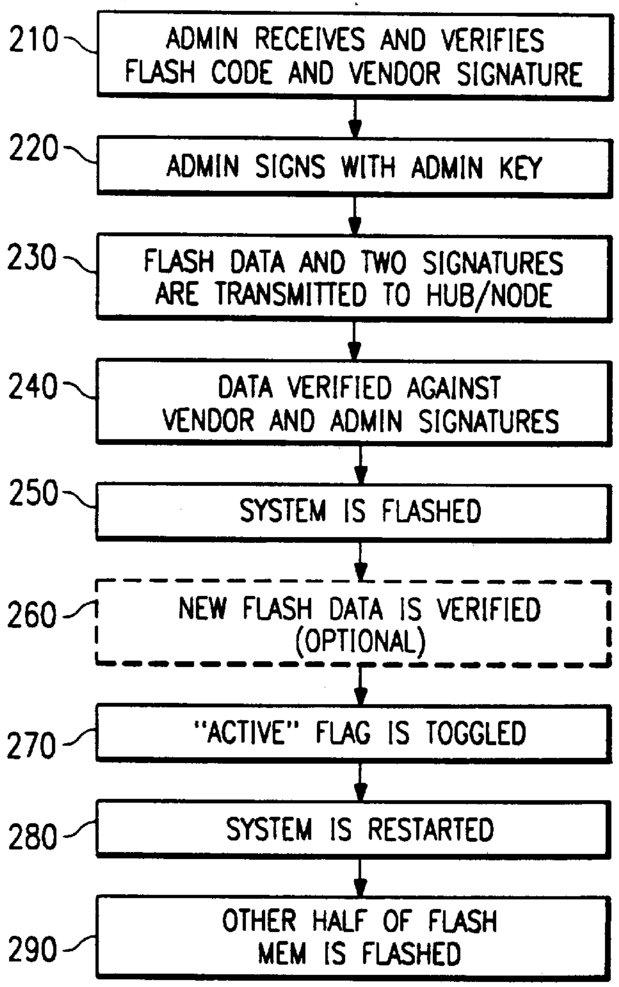

210
ADMIN RECEIVES AND VERIFIES FLASH CODE AND VENDOR SIGNATURE
220
ADMIN SIGNS WITH ADMIN KEY
230
FLASH DATA AND TWO SIGNATURES ARE TRANSMITTED TO HUB/NODE
240
DATA VERIFIED AGAINST VENDOR AND ADMIN SIGNATURES
250
SYSTEM IS FLASHED
260
NEW FLASH DATA IS VERIFIED (OPTIONAL)
270
"ACTIVE" FLAG IS TOGGLED
280
SYSTEM IS RESTARTED
290
OTHER HALF OF FLASH MEM IS FLASHED

FIG. 3

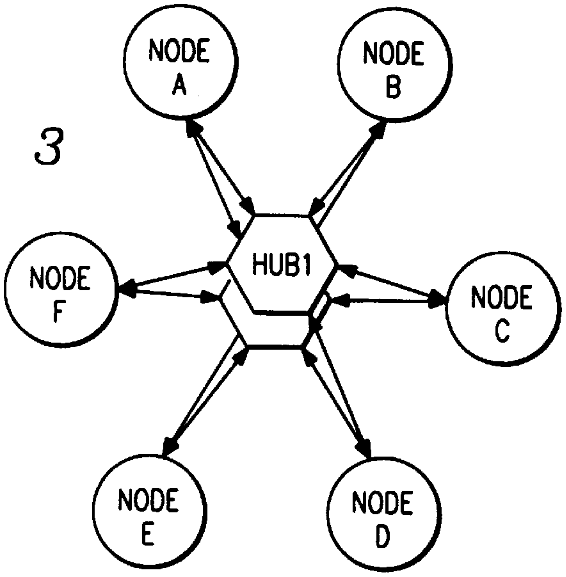


FIG. 4

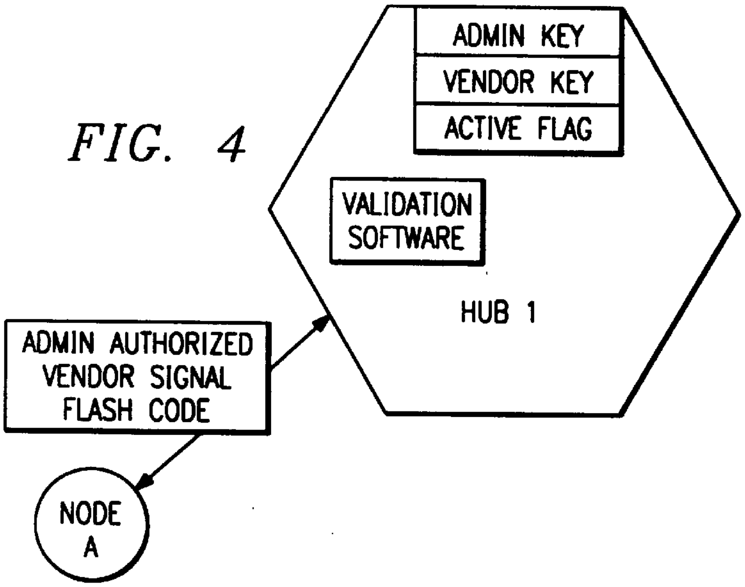
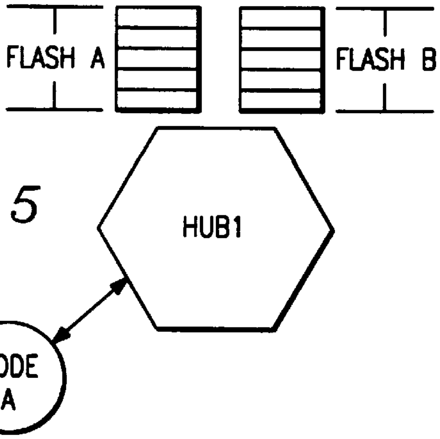


FIG. 5



METHOD FOR THE SECURE REMOTE FLASHING OF A BIOS MEMORY

BACKGROUND AND SUMMARY OF THE INVENTION

This application relates to computer network systems, and more particularly to network flash BIOS memories.

Background: BIOS Updates

A Basic Input/Output System (BIOS) memory is a memory (typically small) which stores the basic software to provide for initial system setup and configuration, and allows the system to load and execute subsequent programs. This configuration software must be available to the system when it is first started, so the BIOS memory must be non-volatile.

In some systems it is sufficient to supply a read-only memory which is hard coded with the BIOS system. With today's rapidly changing technologies, however, it has become advantageous to provide rewritable BIOS memories, so that the BIOS software can be upgraded when necessary. Therefore, many of today's systems use flash or EEPROM memories to store the BIOS software, and provide means for the user of the system to reprogram the BIOS memory when necessary. With a flash BIOS, the BIOS image or a portion of the BIOS image can be updated by a software update. This is often performed by downloading or storing the new software, or "flash" information, onto a media storage device, such as a floppy disk, and executing a program to write the new software into the BIOS memory. This procedure is commonly referred to as "flashing" the memory.

A flash BIOS typically consists of two separately programmable portions, each of which, during normal operation, contains an identical copy of the BIOS software. An "active flag" indicates which memory portion is actually executed when the system is started.

To upgrade a BIOS in flash memory, only half the memory is updated at one time. In order to update the BIOS without ever losing operability, the inactive half of the BIOS, according to the active flag, is overwritten, and then the flag state is changed to make the inactive half active, and then the system is power cycled.

This causes the system to come up in the active side of the BIOS.

Background: Networked Systems

In many common applications, the BIOS must be flashed locally, requiring the operator's actual presence at the machine to be updated. In other systems, the BIOS may be updated remotely, by sending the BIOS upgrade over a telephone connection or local network. Remote flashing makes system upgrades much more convenient, but introduces possible security problems, in that the BIOS may possibly be replaced or corrupted by a remote user or even a "virus" running on a remote system.

In a typical computer network, multiple computer systems are each connected to a node of a common network hub. Typically, one of these nodes is designated as an "admin" node, i.e., a node from which a system administrator can perform remote updating of the BIOS software of the network hub.

On a current networked computer system, each node of a computer network is a computer which may have an individual flash memory, and the network hub also has its own flash memory. In current systems, there is no procedure or system by which to perform upgrades to the BIOS software in a way that can conveniently be handled by system administrators, and which also assures security and compatibility.

Background: Public Key Cryptosystems

In public key cryptosystems, each user has two related complementary keys, a publicly revealed key and a private key. Each key unlocks the code that the other key makes. Knowing the public key does not help you deduce the corresponding private key. The public key can be published and widely disseminated across a communications network. In the context of this application, a public key may be stored in an otherwise vulnerable memory, but that public key is only useful to decrypt data which was encrypted with the corresponding private key.

Background: Digital Signatures

Digital signatures are used to provide message authentication. The sender, for example a software vendor or system administrator, uses his own private key to encrypt a "message digest," thereby 'signing' the message. A message digest is a cryptographically-strong one-way hash function. It is somewhat analogous to a "checksurn" or Cylic Redundancy Check (CRC) error checking code, in that it compactly represents the message and is used to detect changes in the message. Unlike a CRC, however, it is computationally infeasible for an attacker to devise a substitute message that would produce an identical message digest. The message digest gets encrypted by the sender's private key, creating a digital signature of the message. Various digital signature standards have been proposed, such as Secure Hash Algorithm (SHA) or Message Digest **5** (MD**5**)

The recipient can verify the digital signature by using the sender's public key to decrypt it. This proves that the sender was the true originator of the message, and that the message has not been subsequently altered by anyone else, because the sender alone possesses the private key that made that digital signature. Forgery of a signed message is infeasible, and the sender cannot later disavow his digital signature.

These two processes (encryption and digital signatures) can be combined to provide both privacy and authentication by first signing a message with the sender's private key, then encrypting the signed message with the recipient's public key. The recipient reverses these steps by first decrypting the message with his own private key, then checking the enclosed digital signature with the sender's public key. In this way, the encrypted message cannot be read by anyone but the recipient, and it can only have been created by the sender.

Further background on digital signatures can be found, for example, in the following books, all of which are hereby incorporated by reference: Pfitzman, Digital Signature Schemes (1996); Grant, Understanding Digital Signature (1997).

Improved System and Method for FLASH BIOS Upgrades

The present application discloses an improved system and method for FLASH BIOS upgrades which is particularly useful in network hubs. Each hub or node which is equipped with a FLASH memory is also equipped with a validation system, which ensures that a received FLASH upgrade is authorized and uncorrupted. Each set of instructions to be flashed is marked both with a vendor authorization digital signature and also a system administrator authorization digital signature, and both digital signatures must be recognized by the validation system before the FLASH memory will be upgraded. Because digital signatures are used for security purposes, flash upgrades can be performed from any location on the network, and are not limited to an admin node.

BRIEF DESCRIPTION OF THE DRAWING

The disclosed inventions will be described with reference to the accompanying drawings, which show important sample embodiments of the invention and which are incorporated in the specification hereof by reference, wherein:

FIG. **1** shows a block diagram of a computer system with FLASH memory according to the presently preferred embodiment.

FIG. **2** shows a flowchart of the process of the presently preferred embodiment.

FIG. **3** shows a block diagram of a computer network system according to the presently preferred embodiment.

FIG. **4** shows a block diagram of a computer system connected to a network hub according to presently preferred embodiment.

FIG. **5** shows a block diagram of a computer system (node) and hub where the dual-flash variant is employed.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The numerous innovative teachings of the present application will be described with particular reference to the presently preferred embodiment. However, it should be understood that this class of embodiments provides only a few examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily delimit any of the various claimed inventions. Moreover, some statements may apply to some inventive features but not to others.

FIG. **4** shows a block diagram of a computer system NODE A connected to a network hub HUB**1** according to presently preferred embodiment. Node A may be a typical desktop computer or perhaps a network server computer. In this single-flash scenario, Node A obtains flash information from HUB**1**. The Admin in NODE A verifies the flash information and digitally signs. This provides validation for the flash as well as authorization. Next, the Admin transmits a double-signed code to HUB**1**. HUB**1** validates that the code was authorized by Admin and is valid as from the vendor.

FIG. **5** shows a block diagram of a computer system (node) and hub where the dual-flash embodiment is employed. HUB**1** determines which portion of Flash memory, A or B, is not in use. This can be done by looking at a particular bit that indicates which Flash is running. Assuming Flash A is active, HUB**1** then flashes Flash B. HUB**1** revalidates Flash B and sets Flash B active. Finally, HUB**1** generates a reset to boot to the new flash firmware update.

Error conditions returned to the user include the following: if the flash code is not authorized, the system ignores the flash; if the flash code doesn't have a valid vendor digital signature, again, the system ignores the flash; the flash code is determined to be invalid if the flash reboot process times out—HUB**1** then sets Flash A as active, generates a reset, and reboots.

FIG. **2** shows a flowchart of the process of the presently preferred embodiment covering both the single and dual-flash scenarios. In a BIOS upgrade process according to preferred embodiment, the system administrator obtains the upgrade software and loads it into the admin node. The administrator will examine the software, using the vendor's public key, which is also stored in each flash memory, and verify that it contains an appropriate vendor digital signature (step **210**). Having verified the upgrade, the administrator then attaches his own authorization digital signature (step **220**), using his private key, to verify that the software is to be flashed to the target memory. The upgrade software, with both vendor and authorization digital signatures, is transmitted to the target system (step **230**), which may be the network hub or another computer system on the network.

When the target system receives the transmission, it verifies each of the digital signatures (step **240**), using the stored public keys, to ensure that the upgrade is valid and authorized. If it is, the target system then applies the upgrade to the inactive portion of the flash memory (step **250**). The target system may then optionally perform a checksum operation on the inactive memory portion (step **260**) to ensure that it has been properly programmed. The active flag is then toggled to set the newly programmed portion of the memory active (step **270**).

The target system is restarted (step **280**), and the BIOS software is loaded. Since the active flag has been reset, the upgraded software is executed. If it executes with no errors, the other flash memory portion, now inactive, is flashed with the upgrade software (step **290**). If, however, the new software causes the system to crash, or causes some other error, the active flag is automatically toggled back to the memory portion with the known good software (step **270**).

Each set of update software must have two digital signatures. One of these digital signatures identifies the software vendor, which will ensure that only an authentic BIOS upgrade is applied. The second digital signature is an authorization digital signature of the system administrator, which ensures that only authorized BIOS upgrades are applied. The digital signatures can be defined by any convenient digital signature standard (an RSA—Rivest, Shamir, & Adleman—standard is preferred).

The flash memory must therefore contain two public decryption keys according to a dual-key encryption system. One public key will correspond to a private key known only to the vendor, and the other public key will correspond to a private key known only to the system administrator. By using a dual-key digital signature standard, there is little chance of a security compromise if the flash is examined (by users or intruders) to determine the stored public keys. According to the disclosed process, these public keys will still not allow an unauthorized software upgrade to be applied.

The verification of the digital signatures on the target system may be accomplished in multiple ways. The preferred embodiment requires a double digital signature: flash updates must be signed both by the vendor and by the system administrator, and both digital signatures must be identified by the data in the flash memory. This identification is accomplished by a dual-key digital-signature-verification system. "Public" keys for both the vendor and the administrator are stored or hard-coded into the flash memory. The corresponding "private" keys are held by the vendor and system administrator, and only these private keys can generate the digital signatures which the hub can recognize using the public keys stored in the flash memory.

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.