US005844986A

United States Patent [19]

Davis

[11] Patent Number: 5,844,986

[45] Date of Patent: Dec. 1, 1998

[54] **SECURE BIOS**

[75] Inventor: **Derek L. Davis**, Phoenix, Ariz.

[73] Assignee: **Intel Corporation**, Santa Clara, Calif.

[21] Appl. No.: **724,176**

[22] Filed: **Sep. 30, 1996**

[51] **Int. Cl.**[6] **H04L 9/00**

[52] **U.S. Cl.** **380/4**; 380/25

[58] **Field of Search** 380/23, 25, 3, 380/4, 49

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,022,077	6/1991	Bealkowski et al. .	
5,144,659	9/1992	Jones .	
5,289,540	2/1994	Jones	380/4
5,359,659	10/1994	Rosenthal .	
5,377,264	12/1994	Lee et al. .	
5,386,469	1/1996	Yearsley et al.	380/3
5,421,006	5/1996	Jablon .	
5,444,850	8/1995	Chang	380/23
5,450,489	9/1995	Ostrover et al. .	
5,465,299	11/1995	Matsumoto et al. .	
5,479,509	12/1995	Ugon .	
5,568,552	10/1996	Davis	380/4
5,584,023	12/1996	Hsu .	
5,644,636	7/1997	Fernandez	380/4
5,666,411	9/1997	McCarty .	

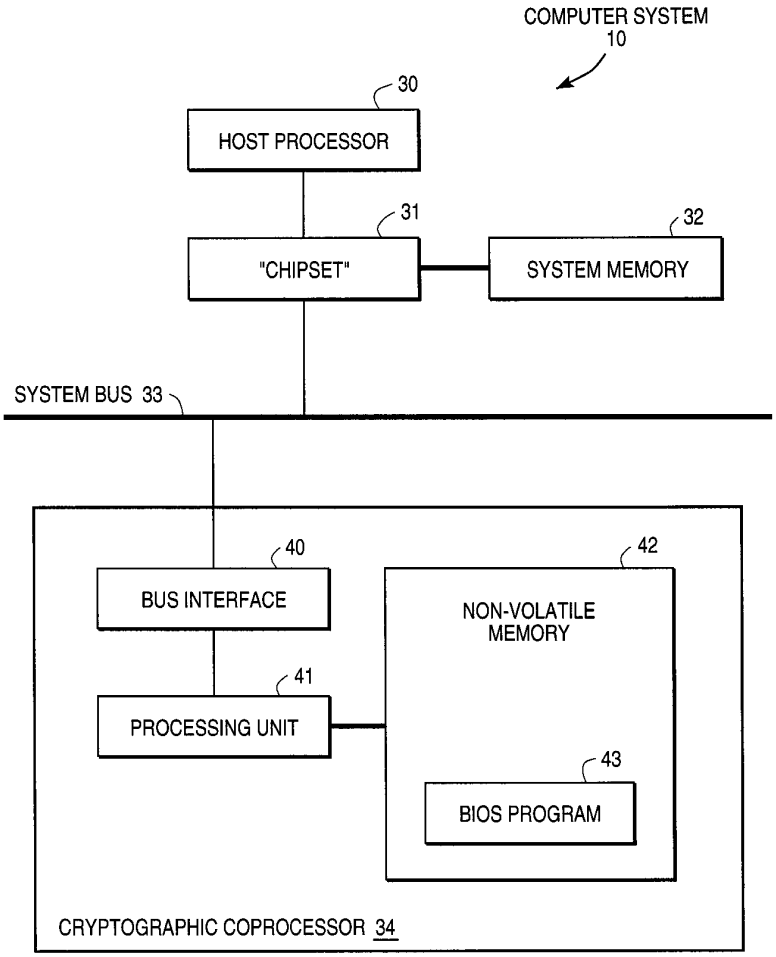
Primary Examiner—David Cain

Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman

[57] **ABSTRACT**

A subsystem prevents unauthorized modification of BIOS program code embedded in modifiable non-volatile memory devices such as flash memory. A cryptographic coprocessor containing the BIOS memory device performs authentication and validation on the BIOS upgrade based on a public/private key protocol. The authentication is performed by verifying the digital signature embedded in the BIOS upgrade.

43 Claims, 3 Drawing Sheets



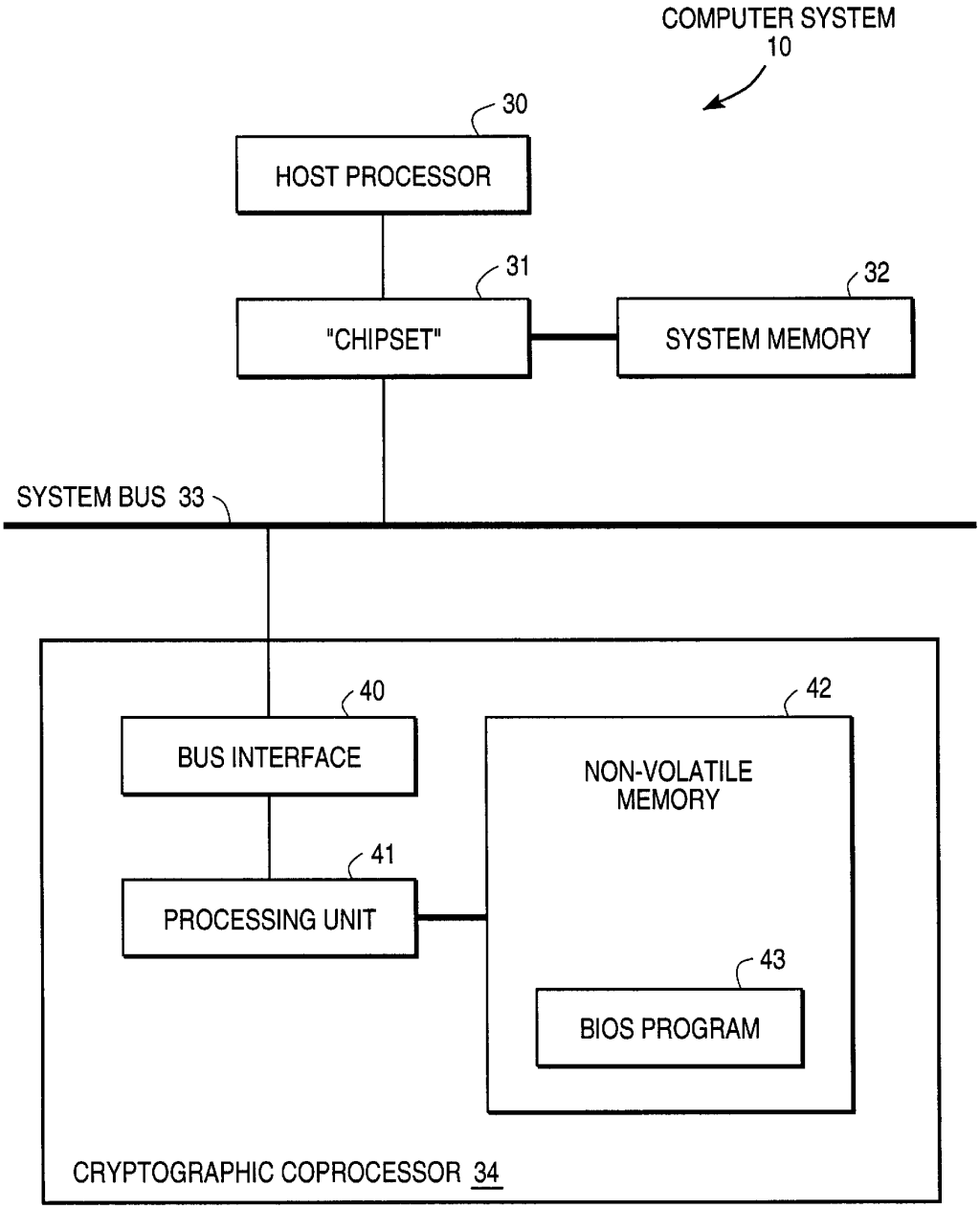


FIG. 1

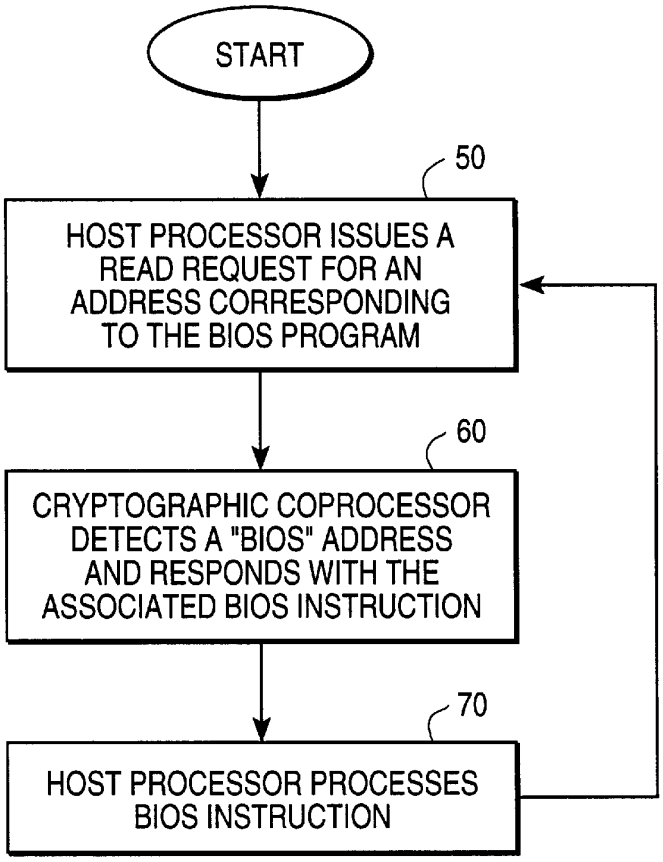


FIG. 2

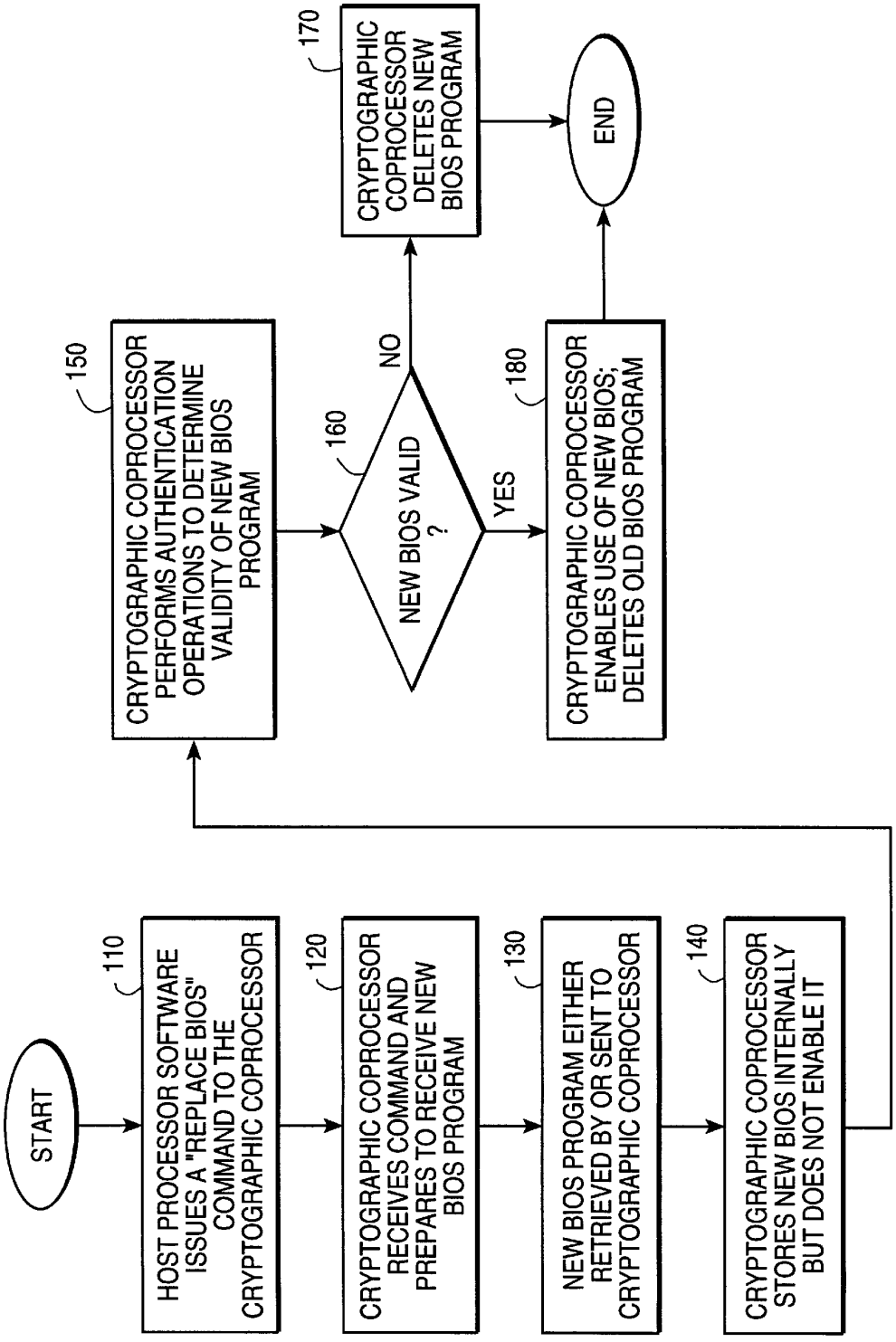


FIG. 3

SECURE BIOS

BACKGROUND OF THE INVENTION

1 . Field of the Invention

This invention relates to the field of security of computer firmware, especially in the areas of Basic Input and Output System ("BIOS") in general computing systems, such as personal computers ("PCs").

2. Description of Related Art

One of the most critical elements in a computer system is the boot-up firmware, such as the Basic Input and Output System ("BIOS"). Typically stored in some form of non-volatile memory, the BIOS is machine code, usually part of an Operating System ("OS"), which allows the Central Processing Unit ("CPU") to perform tasks such as initialization, diagnostics, loading the operating system kernel from mass storage, and routine input/output ("I/O") functions.

Upon power up, the CPU will "boot up" by fetching the instruction code residing in the BIOS. Due to its inherent nature, the BIOS has two conflicting requirements: (1) it should be well protected because if it is modified or destroyed, the entire system will fail, (2) it should be easily modifiable to allow field upgrade for feature enhancement or removal of software bugs.

Traditionally, BIOS is implemented in Erasable Programmable Read Only Memory ("EPROM"). EPROM has an advantage of not being modified in circuit. To modify the contents of the EPROM, the device must be first erased by being removed from the socket and exposed to Ultraviolet light for a prolonged period of time. In this respect, BIOS implemented in EPROM is resistant to virus attack and other electronic sabotages. However, EPROM devices do not support "field upgrades" because these devices are not in-circuit programmable, which is a necessary characteristic for field upgrades. Field upgrading allows customers to upgrade the BIOS in the field to avoid costly delay and parts exchanges. Because of the importance for field upgrading, virtually all BIOS firmware is now implemented using flash memories. However, being field modifiable, BIOS flash memories are vulnerable to virus attacks which could cause devastating results in sensitive applications such as financial transactions.

With no security protection, conventional computer architectures implemented with BIOS flash memories are vulnerable to many kinds of intrusive attacks, such as a virus attack. In a typical virus attack, the virus code executes a code sequence to modify the BIOS flash memory. The code in BIOS flash memory, having no protection, is corrupted and the destructive effects may become effective immediately, when the system is booted up the next time, or when certain conditions or events have occurred. The infected code may further propagate to other areas of the BIOS code or the operating system kernel. Because the BIOS is the first program code to execute when the computer system is "powered up", prior to any system or network virus scanning software, detection and eradication of a BIOS-based virus is extremely difficult. The BIOS-based virus can "hide its tracks" from such scanning software, effectively becoming invisible.

The primary focus of the present invention, therefore, is to prevent corrupting the BIOS by a computer virus. This is achieved by imposing an authentication and validation procedure before the contents of the BIOS flash memory are modified.

The approach which is pursued in this invention builds on the concept of BIOS authentication by incorporating the BIOS flash memories into existing hardware with authenticating capability such as the cryptographic coprocessor. Since the cryptographic coprocessor both stores the BIOS and enforces authentication of BIOS updates, an attacker has no means by which to corrupt the BIOS contents.

SUMMARY OF THE INVENTION

The present invention describes a system to securely update an executable code. The system comprises of a first storage element for storing a code update, a second storage element for storing the executable code that needs to be updated, an identification code for identifying the first storage element and the code update, and a security processor. The security processor is coupled to the second storage element to authenticate and validate the first storage element and the code update using the device identification.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

FIG. **1** is a diagram of the present invention where the BIOS flash memory resides inside a cryptographic coprocessor which may be interfaced to the PCI bus.

FIG. **2** is a flowchart of the operations that occur in the present invention during a normal read access to the BIOS program by the host processor.

FIG. **3** is a flowchart of the operations that occur in the present invention during a field upgrade of the BIOS program.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention provides a procedure to authenticate and validate a code update, such as a BIOS upgrade for example, using cryptographic technology. In the following description, some terminology is used to discuss certain cryptographic features. A "key" is an encoding and/or decoding parameter used by conventional cryptographic algorithms such as Rivest, Shamir and Adleman ("RSA"), Data Encryption Algorithm ("DEA") as specified in Data Encryption Standard ("DES") and the like. A "certificate" is defined as any digital information (typically a public key) associated with an entity, encrypted by a private key held by another entity such as a manufacturer or a widely published trusted authority (e.g., bank, governmental entity, trade association, etc.). A "digital signature" is similar to a certificate but is typically used for authenticating data. Herein, the term "secure" indicates that it is computationally infeasible for an interloper to successfully perpetuate fraud on a system. A security processor is an electronic device capable of performing security functions to provide security protection for the system.

The authentication and validation are performed by a security processor which contains the BIOS firmware. One example of such a security processor is a cryptographic coprocessor. The cryptographic processor authenticates and validates the BIOS firmware by using secret information such as a digital signature embedded in the BIOS upgrade.

Referring to FIG. **1**, an embodiment of a computer system implemented within the present invention is shown. The computer system **10** includes a chipset **31** which operates as an interface to support communications between host pro-

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.