US005802592A

United States Patent [19]

Chess et al.

[11] Patent Number: 5,802,592

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

[54] **SYSTEM AND METHOD FOR PROTECTING INTEGRITY OF ALTERABLE ROM USING DIGITAL SIGNATURES**

[75] Inventors: **David M. Chess**, Mohegan Lake; **Gregory Bret Sorkin**; **Steve Richard White**, both of New York, all of N.Y.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **656,626**

[22] Filed: **May 31, 1996**

[51] **Int. Cl.**[6] **G06F 12/14**; G06F 12/16; G06F 11/30

[52] **U.S. Cl.** **711/164**; 711/102; 711/103; 395/183.12; 395/183.14; 395/183.21; 395/652; 395/633

[58] **Field of Search** 711/102, 163, 711/103, 164; 395/651–653, 186, 188.01, 183.09, 183.12, 183.14, 183.21

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,327,531	7/1994	Bealkowski et al.	395/182.04
5,379,342	1/1995	Arnold et al.	380/2
5,396,558	3/1995	Ishiguro et al.	380/25
5,522,076	5/1996	Dewa et al.	395/652
5,579,522	11/1996	Christeson et al.	395/652
5,634,079	5/1997	Buxton	395/892

FOREIGN PATENT DOCUMENTS

0 515 760 A1	12/1992	European Pat. Off. .
0 588 339 A2	3/1994	European Pat. Off. .

OTHER PUBLICATIONS

Aarons et al., Security strategies: hardware protection for PCs, PC Magazine, v6, p. 104(12), Apr. 28, 1987.

Rosch, Internal Security: The Growing Mass of Stored PC Data Makes Protecting It a Modern Necessity, PC Week, v2, n18, pp. 89–91, May 7, 1985.

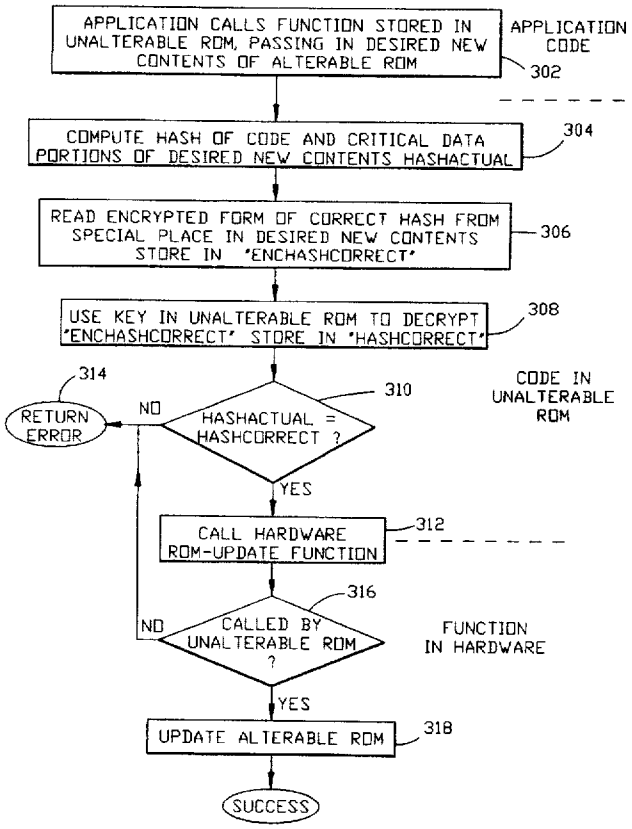
Clark et al., BITS: A smartcard protected operating system, Communications of the ACM, v37, n11, pp. 66–70, Nov. 1994.

Primary Examiner—Eddie P. Chan
Assistant Examiner—Reginald G. Bragdon
Attorney, Agent, or Firm—Perman & Green, LLP

[57] **ABSTRACT**

A system and method for verifying the integrity of a computer system's BIOS programs stored in alterable read only memory (such as FLASH ROM), and preventing malicious alteration thereof. The system and method regularly check the contents of the alterable read only memory using a digital signature encrypted by means of an asymmetrical key cryptosystem.

28 Claims, 3 Drawing Sheets



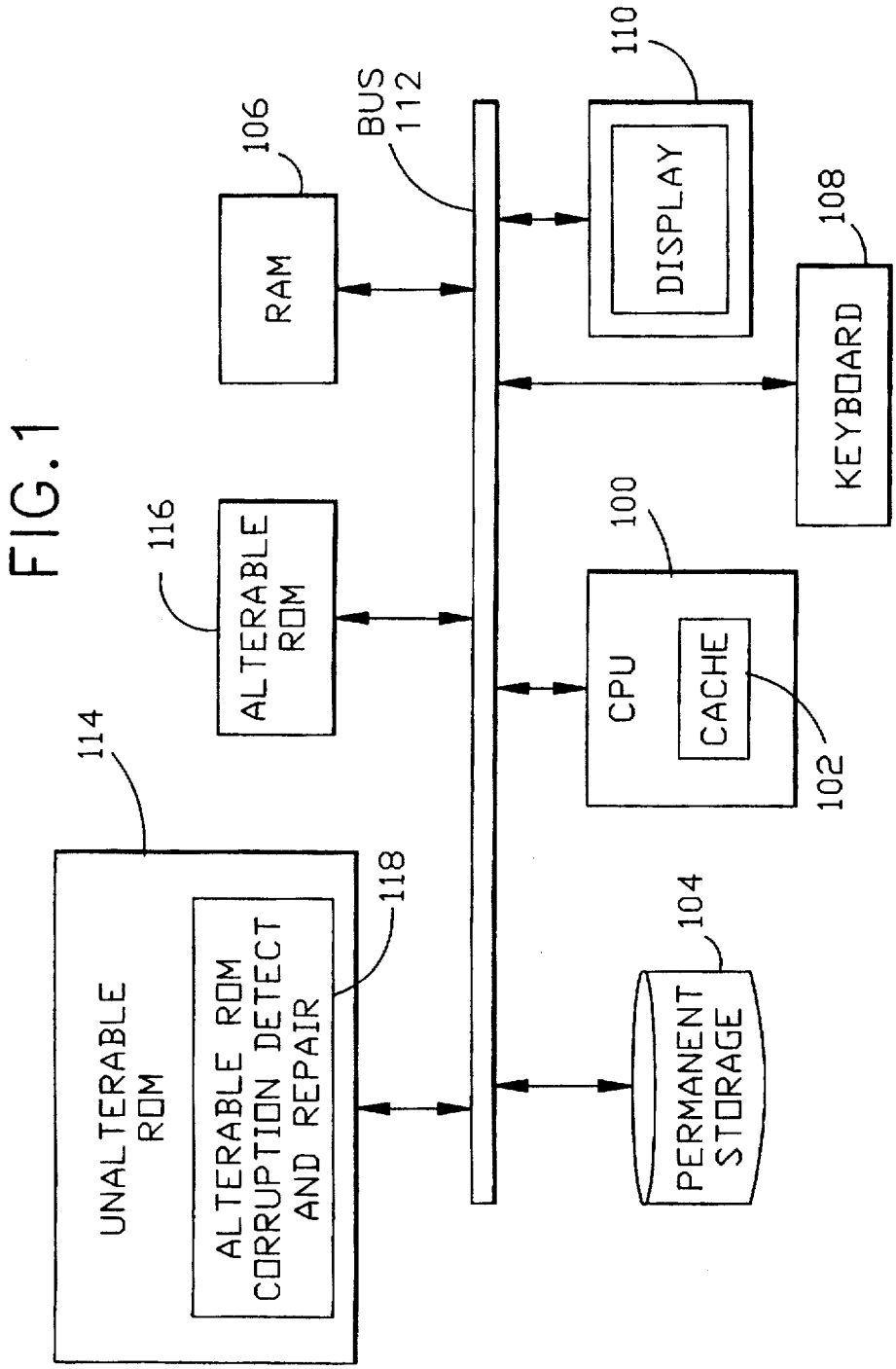


FIG. 1

FIG.2

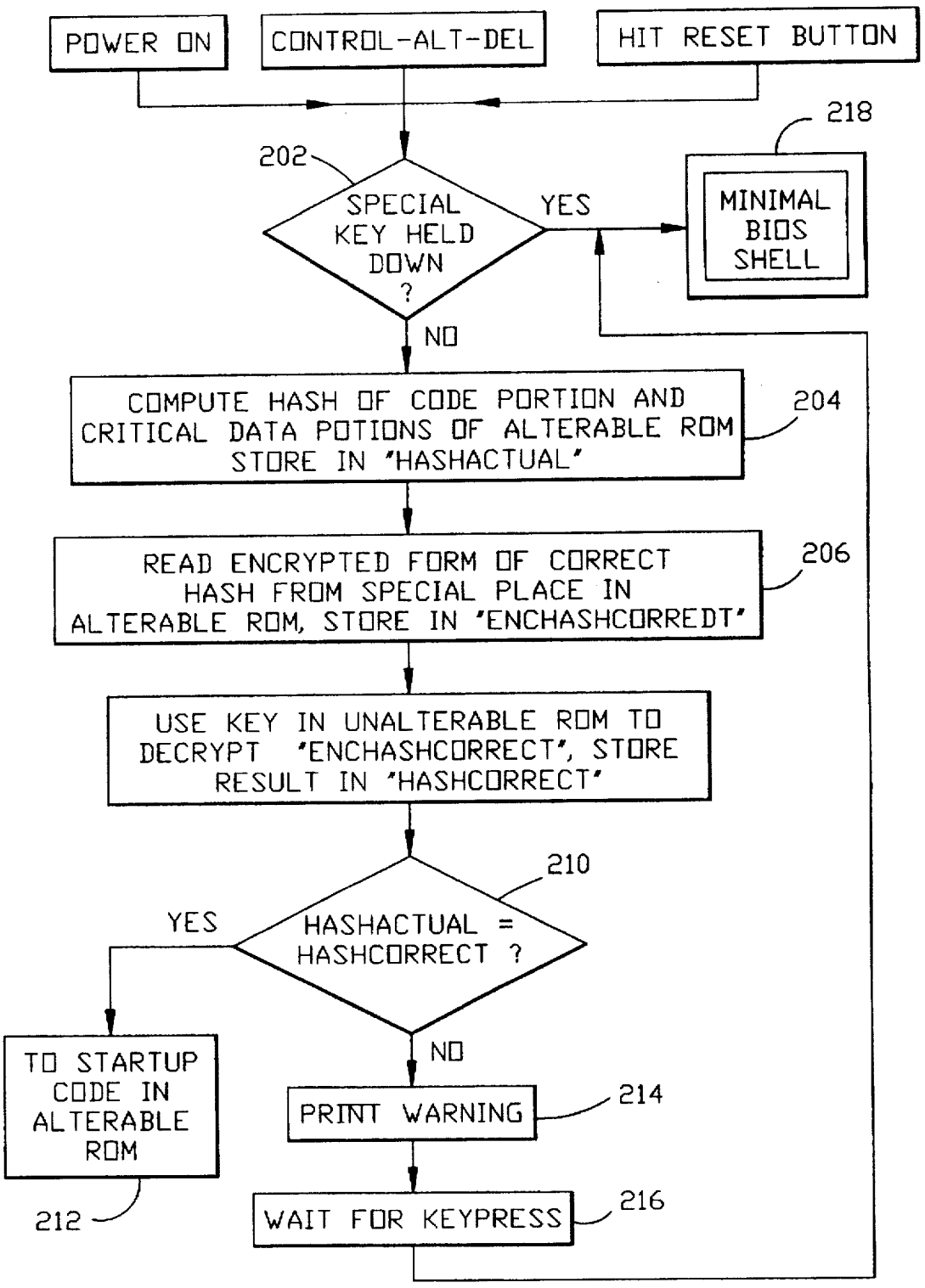
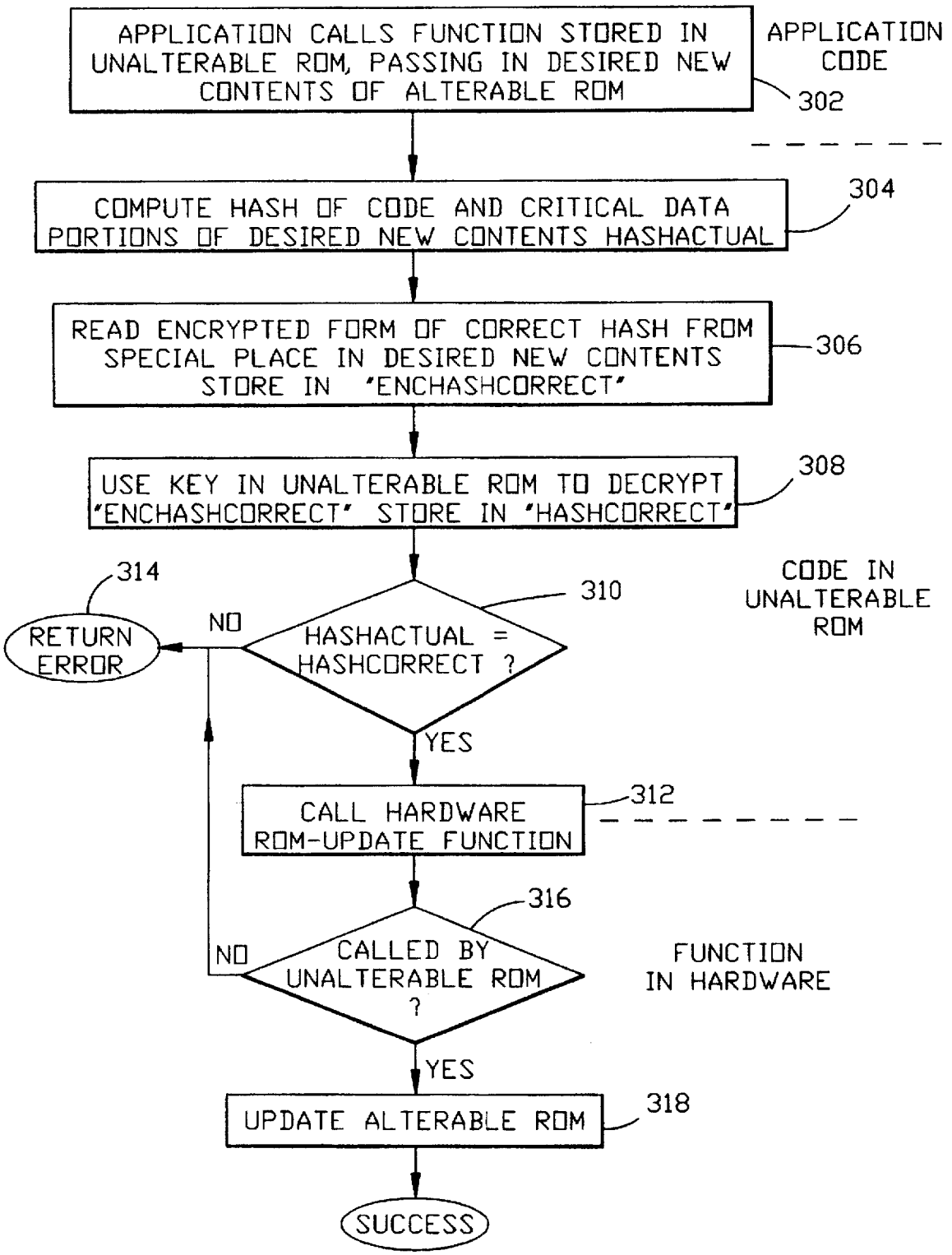


FIG.3



SYSTEM AND METHOD FOR PROTECTING INTEGRITY OF ALTERABLE ROM USING DIGITAL SIGNATURES

FIELD OF THE INVENTION

The invention relates to the protection of the integrity of computer system basic input-output systems.

BACKGROUND OF THE INVENTION

Modern general-purpose computers contain programs stored in non-volatile read-only memory (ROM) which are used to "bootstrap" the system when power is turned on, and to provide basic low-level access to the hardware. These programs generally perform various tests for proper functioning of the system hardware at power-on and then locate, load and transfer control to the operating system bootstrap code. They also provide a standard interface (sometimes called the basic Input/Output System, or BIOS) to the functions of the hardware.

While such system ROMs were originally of the permanently "burned-in" variety, which can be changed only by physically replacing a microchip, advances in technology have recently made it possible to utilize alterable, or "FLASH" ROM instead. The advantage of alterable ROM is that its contents can be altered by software, making ROM updates significantly simpler. As alterable ROM technology advances, and as systems become more complex, requiring more frequent ROM updates, the use of FLASH for this purpose is quickly becoming more common.

While software-alterable ROM has definite advantages, it also has dangers; since the ROM is the basic software that controls the startup and low-level operation of the system, if it becomes corrupted (accidently or maliciously), the integrity of the system as a whole can be compromised, and it can be very difficult either to detect the corruption or to repair it.

There are well-known methods of verifying the integrity of the contents of ROMs (FLASH and otherwise) by performing a simple checksum, to ensure that, to a very high probability, no accidental changes have been made to the contents of the ROM. The techniques used to do this verification are typically a simple additive checksum or a cyclic redundancy check: these techniques are designed to be simple and fast, while having a high probability of detecting typical accidental or defect-caused changes to ROM. They are, however, easily "invertible" ; that is, given the current contents of ROM and the current value of the checksum, an attacker desiring to make intentional changes to the ROM without modifying the checksum would be able to do so with little difficulty.

A further feature of many current systems is that they allow the user to access the built-in programs stored in ROM for examining and altering system configuration settings. This typically is accomplished by starting the system from a special diskette, or pressing a combination of keys during system setup. But the configuration programs, and the programs that decide whether or not to pass control to them, are themselves alterable ROM (on machines that have alterable ROM), and therefore could become corrupted.

SUMMARY OF THE INVENTION

The current invention functions in a component of a computing system containing alterable ROM to verify that the alterable ROM has not been changed, or that a proposed update to the alterable ROM is legitimate. This verification is performed by use of a digital signature, the signature having the characteristic that it is not easily invertible: even an attacker with full knowledge of the code used to verify the digital signature, and with the ability to alter the current contents of the ROM and the current signature, would have to perform a prohibitive amount of computation to generate a new content/signature pair that would pass the test.

The manufacturer, on the other hand, by virtue of having access to a secret piece of data (for example, the private key in an asymmetrical-key cryptosystem), is able to produce signatures for new versions of the contents of alterable ROM very easily.

BRIEF DESCRIPTION OF THE DRAWING

FIG. **1** is a system in accordance with the invention.

FIG. **2** is a flow diagram describing a method for checking the integrity of an alterable ROM, in accordance with one aspect of the invention.

FIG. **3** is a flow diagram describing a method for updating or restoring the contents of an alterable ROM in accordance with a further aspect of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. **1** is a block diagram of a system in accordance with the present invention.

The system includes a CPU **100** with optional cache memory **102**, a permanent storage device **104**, such as a hard disk drive, random access memory **106**, an input device such as keyboard **108**, and an output device, such as display **110**. The system components are connected via bus **112**.

The system further comprises an unalterable ROM **114**, which stores various programs used to bootstrap the system at startup and provide basic low level system hardware access. Also provided is an alterable ROM **116**, such as a FLASH ROM, which stores additional bootstrapping and hardware access programs. The programs in the ROMs **114** and **116** together constitute first and second portions of a general bootstrap program.

Also provided in accordance with the invention is an alterable ROM corruption detect and repair means **118**. The means **118** can be implemented as software running in unalterable ROM **114**. Means **118** operates as described with respect to FIGS. **2** and **3** to detect unauthorized modifications to the alterable ROM **116**, and also either to restore the alterable ROM to its uncorrupted state, or to make authorized changes to the alterable ROM. Means **118** can either constitute part of unalterable ROM **114**, or reside in a separate hardware or software location in the system.

In one embodiment of the invention, a system bootstrap routine is stored in unalterable ROM **114**, the routine performing, when called, a signature computation on the current contents of the alterable ROM **116** and the current signature (stored in ROM **114** or elsewhere), and then passes control to the bootstrap code in the ROM **116** only if the signature is validated. Defect-caused or malicious changes to the FLASH ROM would therefore prevent the system from starting up correctly at the next power-on. The system could also be configured so that an attempt to update ROM **116** will cause an immediate restart from unalterable ROM **114**, immediately revealing a corrupted update. More complex implementations involve a secure-update module that guarantees (for example by monitoring instruction fetches using methods known to the art) that ROM **116** updates could be done only by code running from the ROM **116** itself, and each version of the alterable ROM could contain

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.