SOFTWARE PROTECTION AGAINST FAULT ATTACKS

**Abstract:** A method for protecting information in a device includes providing a device with a non-secure hardware domain, a processor having a software-controlled mode of operation, and a secure hardware domain having a secure memory that is inaccessible by the processor when the processor is operating in the software-controlled mode of operation. Data from the non-secure hardware domain is established in the secure hardware domain. Computing operations are executed on the data in the secure hardware domain to produce a result. The secure hardware domain is purged, while retaining the result therein. The result is thereafter returned from the secure hardware domain into the non-secure hardware domain.

**Diagram:**

![Diagram of secure and non-secure domains](image-url)
SOFTWARE PROTECTION AGAINST FAULT ATTACKS

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

This invention relates to securing data on a computing device. More particularly, this invention relates to prevention of fault attacks that could lead to unauthorized access to information or information protection features on a computing device.

DESCRIPTION OF THE RELATED ART

Embedded security refers to security features built into a device, including physical tamper-resistance features, cryptographic keys and algorithms. Embedded security features can be found today on a variety of computing devices, e.g., personal computers and servers, cellular telephones, set-top boxes, and many appliances. The present invention is largely concerned with protection of data generally, and cryptographic keys in particular. The meanings of several acronyms used in this disclosure are given in Table 1.

Table 1 - Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DH</td>
<td>Diffie-Hellman</td>
</tr>
<tr>
<td>ECC</td>
<td>Elliptic Curve Cryptography</td>
</tr>
<tr>
<td>ECDH</td>
<td>Elliptic Curve Diffie-Hellman</td>
</tr>
<tr>
<td>ECDSS</td>
<td>Elliptic Curve Algorithm with Digital Signature Standard</td>
</tr>
<tr>
<td>MPU</td>
<td>Memory Protection Unit</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest, Shamir, &amp; Adleman</td>
</tr>
</tbody>
</table>
SUMMARY OF THE INVENTION

The above-noted and other computing devices that process encrypted data are potentially compromised by fault attacks, which are defined in further detail below. These attacks are active forms of side channel attacks that involve creation of some fault during the operation of the device being attacked and observation of the result. The faults result in an erroneous computation. Should the computation involve the use of a cryptographic key, comparison between the correct and flawed data may allow information about the key to be extracted. Alternatively, the analysis of the difference in behavior between the flawed device and a normal device can be exploited.

Fault attacks can penetrate modern encryption and decryption systems. For example, fault attacks have recovered cryptographic keys from systems using elliptic curve encryption, RSA and AES algorithms.

During the past ten years many types of fault attacks have been devised. They can be created by a variety of physical effects, e.g., heat, radiation, power variations, optical energy, electromagnetic fields, and mechanical disturbances.

According to disclosed embodiments of the invention, methods and systems are provided for coordinating data processing activities that occur in a non-secure hardware domain with cryptographic operations relating to the data processing that occur in a secure hardware domain as a countermeasure to fault attacks. While the cryptographic operations are occurring, the secure hardware domain is isolated from the non-secure hardware domain and its data, including intermediate computations, and its internal states are inaccessible from the non-secure hardware domain and inaccessible to inquiries from external sources.

Prior to returning a result of cryptographic operations from the secure hardware domain to the non-secure hardware domain, memory in the secure hardware domain is purged, except for the result itself, and internal states within the secure hardware domain are reset. Thereafter, the results are returned to the non-secure hardware domain. The cryptographic operations are impervious to fault attacks that could compromise a private cryptographic key.
An embodiment of the invention provides a method for protecting information in a device, which is carried out by providing a device with a non-secure hardware domain having data stored therein and including a software-controlled processor. The device has a secure hardware domain that includes a secure memory that is inaccessible by the processor when the processor is operating under software control. The method is further carried out by establishing the data in the secure hardware domain, executing computing operations on the data in the secure hardware domain to produce a result, purging the secure hardware domain by deleting data while retaining the final result, and thereafter returning the result from the secure hardware domain into the non-secure hardware domain.

According to one aspect of the method, the data includes a cryptographic key, and the computing operations comprise applying the cryptographic key to the data for encryption or decryption thereof.

According to another aspect of the method, purging includes deleting the cryptographic key from the secure hardware domain.

According to yet another aspect of the method, executing computing operations is performed by a hardware accelerator.

In still another aspect of the method, the secure hardware domain includes a random access memory, wherein establishing data includes storing the data therein and the random access memory stores intermediate results of the computing operations, and purging includes deleting the intermediate results.

Other embodiments of the invention provide computer software product and apparatus for carrying out the above-described method.
BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention, reference is made to the detailed description of the invention, by way of example, which is to be read in conjunction with the following drawings, wherein like elements are given like reference numerals, and wherein:

Fig. 1 is a block diagram of a generic data processing system that is constructed and operative in accordance with a disclosed embodiment of the invention;

Fig. 2 is a block diagram of a generic data processing system that is constructed and operative in accordance with an alternative embodiment of the invention; and

Fig. 3 is a flow chart of a method for performing fault attack-resistant cryptographic operations in a secure hardware domain, in accordance with a disclosed embodiment of the invention.
DETAILED DESCRIPTION OF THE INVENTION

In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one skilled in the art, however, that the present invention may be practiced without these specific details. In other instances, well-known circuits, control logic, and the details of computer program instructions for conventional algorithms and processes have not been shown in detail in order not to obscure the present invention unnecessarily.

Software programming code, which embodies aspects of the present invention, is typically maintained in permanent storage, such as a computer readable medium. In a client/server environment, such software programming code may be stored on a client or a server. The software programming code may be embodied on any of a variety of known tangible media for use with a data processing system, such as a diskette, or hard drive, or CD-ROM. The code may be distributed on such media, or may be distributed to users from the memory or storage of one computer system over a network of some type to other computer systems for use by users of such other systems.

OVERVIEW

Fault attacks produce an abnormal condition or defect at a component, system, or sub-system level, which may lead to a failure, improper functionality or data change. Usually these attacks are non-deterministic and limited. For example, in a non-deterministic register fault attack, an attacker is not able to obtain full control over a target device to set register bits, but may be able to change registers randomly.

In a limited fault attack, specific changes can be effected in the target device, but only in a limited manner. For example, the attacker may be unable to change values of a register to a desired state, but may be able to force all bits to "0" or to "1".

Such fault attack may create a change in bits of a device register at run time, for example while data are being unloaded to a user after completion of a cryptographic operation. Under these circumstances, analysis of the results in memory,
together with intermediate calculation values may allow deduction of at least a portion of a private cryptographic key.

**EMBODIMENT 1**

Turning now to the drawings, reference is initially made to Fig. 1, which is a block diagram of a generic data processing system 10 that is constructed and operative in accordance with a disclosed embodiment of the invention. The architecture shown in Fig. 1 is exemplary. Many suitable variations will occur to those skilled in the art.

The system 10 is segmented into an insecure hardware domain 12 for general operations in accordance with the function of the device and a secure hardware domain, in which cryptographic operations occur. In this embodiment of the system 10, the domains 12, 14, may be realized as separate devices 16, 18, which can be linked via any suitable communications channel 26. For example, the device 16 may be a storage device, such as an information storage card, and the device 18 may be a microprocessor that is adapted to servicing the device 16. The devices 16, 18 need not even be physically connected, and can be at any distance from one another, so long as at least intermittent communication is possible in order to transfer data and control signals therebetween.

The device 16 includes a processing element, central processing unit 20, provided with suitable memory for carrying out normal processing functions. External communication in the domain 12 can occur via an I/O facility 24.

General data processing occurs in the domain 12, using the central processing unit 20 as is well known in the art. In the course of such data processing, it is necessary from time to time to decrypt or encrypt data. Private keys, held in a secure, non-volatile memory 22, and the subject data are placed in a secure memory 32. The memory 22 may be implemented as a separate circuit or chip that is incorporated in the secure hardware domain for use in cryptographic operations and verification of data. Cryptographic operations are then performed in the memory 32, optionally under control of a hardware accelerator 34, which can be actuated by the central processing unit 20. The hardware accelerator 34 may be a PKI accelerator that is adapted to known cryptographic algorithms, such as RSA, ECC, AES, and DES.
During cryptographic operations the central processing unit 20 has no access to the memory 32, nor to any internal registers (not shown) of the hardware accelerator 34. Thus, elements of the device 16 comprise the domain 14 and perform cryptographic operations in isolation, and the domain 14 is protected from access by non-trusted software that could exploit faults that may exist during the cryptographic operations. Upon completion of the cryptographic operations, private keys and intermediate computations in the memory 32 are erased, and results 35 are uploaded to the device 18. Details of the cryptographic operations are described below.

EMBODIMENT 2

Reference is now made to Fig. 2, which is a block diagram of a data processing system 36 that is constructed and operative in accordance with an alternative embodiment of the invention. A non-secure hardware domain and a secure hardware domain are realized in a single computing device 38 that holds a central processing unit 40 and a non-volatile memory 42, which is used for storage of private cryptographic keys. Like the memory 22 (Fig. 1), the memory 42 may be implemented as a separate circuit or chip and incorporated in the secure hardware domain. A non-secure memory 43 is provided for general use by the central processing unit 40, including storage of results of cryptographic operations. A secure memory 45 is used for cryptographic operations. A memory protection unit 46 (MPU) is used to prohibit access by the central processing unit 40 to the secure memory 45 during cryptographic operations. The memory protection unit 46 can split the memory into multiple secure or trusted and non-secure or non-trusted domains to enable protection to be applied to a desired secure domain. Execution of cryptographic operations in the secure memory 45 is facilitated by optional hardware accelerator 34, as in Embodiment 1. The hardware accelerator 34 and secure memory 45 constitute a secure hardware domain, protected by the memory protection unit 46, while other elements of the device 38 form a non-secure hardware domain. In order to perform cryptographic operations, private keys are transferred from the memory 42 to the secure memory 45. Encrypted data are placed in the secure memory 45. As in Embodiment 1, and as explained in further detail below, the secure memory 45 is purged prior to transferring calculation results to the non-secure memory 43, which of course remains accessible to the central processing unit 40.
OPERATION

Reference is now made to Fig. 3, which is a flow chart of a method for performing a fault attack-resistant cryptographic operation in a secure hardware domain, in accordance with a disclosed embodiment of the invention. It is assumed that encrypted data and a private cryptographic key are available in a non-secure hardware domain. At initial step 52, an application requires data to be subjected to cryptographic operations. While decryption is presented by way of example, the method is also applicable, mutatis mutandis, to encrypt data.

Control now proceeds to step 54. The private cryptographic key is placed into the secure hardware domain, e.g., uploaded from the non-secure hardware domain to the secure hardware domain. Typically, step 54 is performed using a CPU in the non-secure hardware domain. A fault at this stage would be detected, as decryption would occur using an incorrect key, and the results would be evident in the subsequent program flow.

Step 56 begins after placing or uploading the private key into the secure hardware domain at step 54. Data to be decrypted is established in the secure hardware domain, e.g., by upload to the non-secure hardware domain, or by creating the secure hardware domain by controlling access to the memory holding the data, e.g., using a memory protection unit. A fault, such as a register fault, in the CPU at this stage would not result in revelation of any information concerning the private cryptographic key at this stage. The application may continue execution in other threads or respects without reference to the private cryptographic key while awaiting decryption to complete. Alternatively, the application may simply sleep or otherwise discontinue further progress until decryption is complete. All communication channels that would allow communication of data or control signals between the secure hardware domain and the non-secure hardware domain are now closed.

Next, at step 58, decryption of the data that was the subject of step 56 is performed by applying the private cryptographic key in accordance with the applicable algorithm. Operation in the secure hardware domain is particularly suitable for cryptographic algorithms having relatively long intermediate states, e.g., RSA, DH, DSS, ECDH, ECDSS and other PKI based algorithms. As noted above, step 58 may be
done under the control of a hardware controller or a software module that, at least at this
stage, lacks the ability to move data from the secure hardware domain to the non-secure
hardware domain. In particular, the CPU, and thus the executing application, and
external inquirers have no access to data or internal states within the secure hardware
domain. The controller in the secure hardware domain sets its internal state in
accordance with any parameters received in step 56. A fault at this stage at worst could
produce an incorrect decryption, which would be detected.

After completion of step 58, at step 60 all information concerning the
decryption of data in the memory of the secure hardware domain is purged, except that
the final result is preserved. The purging function is unable to copy data. Similarly, any
internal state registers in the controller are reset, so that their states bear no relation to
the decryption. Any fault that may be present at this point cannot reveal any information
regarding the private cryptographic key.

Next, at step 62, data communication channels between the secure
hardware domain and the non-secure hardware domain are reopened, and the final result
of the cryptographic operation in step 58 is unloaded or placed in the non-secure
hardware domain. This step may involve a physical movement of data between the
domains, or may be accomplished by the re-establishment of access to the data by
elements in the non-secure hardware domain.

At final step 64 the application that required the decrypted data
continues.

It will be appreciated by persons skilled in the art that the present
invention is not limited to what has been particularly shown and described hereinabove.
Rather, the scope of the present invention includes both combinations and sub-
combinations of the various features described hereinabove, as well as variations and
modifications thereof that are not in the prior art, which would occur to persons skilled
in the art upon reading the foregoing description.
CLAIMS

1. A method for protecting information in a device, comprising the steps of:
   providing a device (18) with a non-secure hardware domain (12), a processor (20) having a software-controlled mode of operation, and a secure hardware domain (14) having a secure memory (32) that is inaccessible by said processor when said processor is operating in said software-controlled mode of operation;
   establishing data from the non-secure hardware domain in the secure hardware domain;
   executing computing operations on said data in said secure hardware domain to produce a result;
   purging said secure hardware domain while retaining said result therein;
   and
   thereafter returning said result from said secure hardware domain into said non-secure hardware domain.

2. The method according to claim 1, wherein purging comprises deleting said data from said secure hardware domain.

3. The method according to any of claims 1-2, wherein said data comprises a cryptographic key, wherein said computing operations comprise applying said cryptographic key to said data for encryption or decryption thereof.

4. The method according to claim 3, wherein purging comprise deleting said cryptographic key.

5. The method according to any of claims 1-4, wherein executing computing operations is performed by a hardware accelerator.

6. The method according to any of claims 1-5, wherein said secure hardware domain comprises a random access memory, wherein said step of establishing
data comprises storing said data therein and said random access memory stores intermediate results of said computing operations, and said step of purging comprises deleting said intermediate results.

7. A device comprising:
   a non-secure hardware domain (12);
   a processor (20) having a software-controlled mode of operation;
   a secure hardware domain (14) having a secure memory (32) that is inaccessible by said processor when said processor is operating in said software-controlled mode of operation; and
   a controller operative to upload data to be protected from said non-secure hardware domain into said secure memory, to execute computing operations on said data in said secure hardware domain to produce a result, and to delete information from said secure memory while retaining said result therein, and thereafter to unload said result from said secure hardware domain into said non-secure hardware domain.

8. The device according to claim 7, further comprising a switch for disabling communication between said secure hardware domain and said non-secure hardware domain.

9. The device according to any of claims 7-8, wherein said controller comprises a hardware accelerator.

10. The device according to any of claims 7-9, wherein said controller comprises software instructions embedded in a tangible medium.

11. The device according to any of claims 7-10, wherein said data comprises a private cryptographic key, execution of said computing operations comprises applying said private cryptographic key to others of said data, and deleting said information comprises purging said private cryptographic key from said secure memory.
12. The device according to claim 11, wherein deleting said information further comprises resetting an internal state of said controller.

13. A computer software product for protecting information from malicious action, including a tangible computer-readable medium in which computer program instructions are stored, which instructions, when read by a device (18), cause the device to upload data from a non-secure hardware domain (12) in the device into a secure hardware domain (14) thereof, said non-secure hardware domain including a processor (20) having a software-controlled mode of operation, wherein said secure hardware domain is inaccessible by said processor when said processor is operating in said software-controlled mode of operation, execute computing operations on said data in said secure hardware domain to produce a result, purge said secure hardware domain while retaining said result therein, and thereafter unload said result from said secure hardware domain into said non-secure hardware domain.

14. The computer software product according to claim 13, wherein said device is further instructed to purge said secure hardware domain by deleting said data from said secure hardware domain.

15. The computer software product according to any of claims 13-14, wherein said data comprises a cryptographic key, wherein said computing operations comprise applying said cryptographic key to said data for encryption or decryption thereof.

16. The computer software product according to claim 15, wherein said device is further instructed to purge said secure hardware domain by deleting said cryptographic key.

17. The computer software product according to any of claims 13-16, wherein said device is further instructed to purge said secure hardware domain by deleting intermediate results of said computing operations.
FIG. 3

1. **INITIATE DECRIPTION** 52
2. **UPLOAD PVT KEY** 54
3. **UPLOAD ENCRYPTED DATA** 56
4. **PERFORM DECRYPTION** 58
5. **PURGE SECURE DOMAIN** 60
6. **DOWNLOAD DECRYPTED DATA** 62
7. **RESUME PROCESSING** 64
### A. CLASSIFICATION OF SUBJECT MATTER

**INV. G06F21/02**

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)

**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**

### C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document with indication</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>WO 2005/091109 A (NOKIA CORP [FI]; PAATERO LAURI [FI]) 29 September 2005 (2005-09-29) page 8, line 24 - page 13, line 25</td>
<td>1-17</td>
</tr>
</tbody>
</table>

### D

Further documents are listed in the continuation of Box C

| X | See patent family annex |

- Special categories of cited documents
  - "A" document defining the general state of the art which is not considered to be of particular relevance
  - "E" earlier document but published on or after the international filing date
  - "L" document which may throw doubts on novelty claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  - "O" document referring to an oral disclosure or other means
  - "P" document published prior to the international filing date but later than the priority date claimed

- "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
- "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
- "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
- "S" document member of the same patent family

**Date of the actual completion of the international search**

5 February 2009

**Date of mailing of the international search report**

13/02/2009

**Name and mailing address of the ISA/”**

European Patent Office, P B 5818 Patentlaan 2 NL - 2280 HV Rijswijk

Tel (+31-70) 340-2040, Fax (+31-70) 340-3016

**Authorized officer**

Cartrysse, Kathy
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2005132186 A1</td>
<td>16-06-2005</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>US 2007180539 A1</td>
<td>02-08-2007</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20060127205 A</td>
<td>11-12-2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2005210280 A1</td>
<td>22-09-2005</td>
</tr>
</tbody>
</table>