Computer Science and Application 计算机科学与应用, 2017, 7(4), 310-319
Published Online April 2017 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2017.74038

Operating System Integrity Measurement Method Based on UEFI

Yihua Zhou[1], Hui An[1,2], Guan Wang[1,2], Liang Sun[3]

[1]College of Computer Science, Beijing University of Technology, Beijing
[2]Key Laboratory of Trustworthy Computing in Beijing, Beijing
[3]ZD Technologies (Beijing), Beijing
Email: 1216589268@qq.com

Received: Apr. 2nd, 2017; accepted: Apr. 14th, 2017; published: Apr. 19th, 2017

Abstract

If the operating system kernel is under attack, it can pose a significant threat to operating systems and applications. In order to ensure the integrity of the operating system kernel, this paper presents an operating system integrity measurement method based on UEFI firmware. In the scheme, we measure the integrity of operating system mainly in UEFI BIOS boot process, using TCM chip's encryption, authentication functions and Hash algorithm. The scheme can effectively protect the kernel and the safety of the operating system.

Keywords

UEFI, OS Kernel, TCM, Integrity Measurement

基于UEFI的操作系统完整性度量方法

周艺华[1]，安　会[1,2]，王　冠[1,2]，孙　亮[3]

[1]北京工业大学计算机学院，北京
[2]可信计算北京市重点实验室，北京
[3]中电科技(北京)有限公司，北京
Email: 1216589268@qq.com

收稿日期：2017年4月2日；录用日期：2017年4月14日；发布日期：2017年4月19日

摘　要

操作系统内核受到攻击，会对操作系统以及应用程序造成重大的威胁，为了保证操作系统内核的完整性，

文章引用: 周艺华, 安会, 王冠, 孙亮. 基于 UEFI 的操作系统完整性度量方法[J]. 计算机科学与应用, 2017, 7(4): 310-319. https://doi.org/10.12677/csa.2017.74038

本文提出了一种基于UEFI固件的操作系统完整性度量机制，该方案主要在UEFI BIOS启动过程中，利用TCM芯片的加密，认证和Hash运算等技术，对操作系统内核进行完整性度量，能够有效保护内核以及操作系统的安全。

关键词

UEFI，操作系统内核，TCM，完整性度量

1. 引言

操作系统是计算机硬件和软件的纽带，是应用软件运行的基础环境，而内核的安全性是操作系统安全的核心问题，若操作系统内核受到攻击，会对操作系统以及应用程序造成重大的威胁。

文献[1]中指出了可信验证包括启动验证和运行时验证两种类型，启动验证来源于认证启动[2]，认证启动保证了操作系统开机启动时是安全可信的，如果开机启动时不能够保证操作系统的安全性，则开机启动之后的验证将不再有意义。如何在开机启动时保护操作系统的完整性，防止恶意程序攻击内核系统是可信计算[3] [4]亟待解决的问题。本文通过在 UEFI BIOS 启动的过程中，基于 TCM [5]芯片的加密和认证等功能，完成对操作系统内核文件的完整性度量，从而保证了操作系统内核的完整性和安全性。

IMA [6] (IBM integrity measurement)提出了一种有效的度量方法，该方法是在加载动态链接库或内核模块时，对用到的一些关键数据和信息进行度量，并将度量的结果扩展到通过 TPM 来保护的信任链中。IMA 架构要求任何软件在加载的时候都要被度量，而用户可能只关心某一部分的完整性，对于这些用户来说 IMA 度量会有冗余，会降低系统的效率[2]。

而本文提出的基于 UEFI [7]的操作系统完整性度量方法只是对操作系统的内核进行完整性度量，而且适用于一台计算机装一个或者多个操作系统的情况。该方法是以硬件芯片作为起点，度量在 BIOS 中执行，增加了其自身的安全性，相对于那些在应用层做度量的方法更安全。而且将被检测的对象分离开，更能够有效的防止篡改和不可旁路性，在实现技术上，通过一台安全管理服务器来管理度量初始值的安全性，更简单方便。

UEFI (United Extensible Firmware Interface)是 Intel 联合业界采用开源方式共同制定推出的规范[8]。UEFI BIOS 定义了操作系统和平台固件之间的接口标准，是运行在操作系统和硬件之间的一个新的模型[9]。UEFI 较 BIOS 有可编程性好，可扩展性好，安全性高等优点，所以能够很快速的取代传统 BIOS，成为新一代的趋势。

2. 基于UEFI的操作系统完整性度量的总体设计

基于 UEFI 的操作系统完整性度量主要包括两部分，一部分是需要度量的客户机，另一部分是进行安全管理的服务器。整个系统的流程图如图 1 所示，客户机首先在 UEFI BIOS 启动的时候获取操作系统的内核以及版本，对操作系统内核做 Hash 运算，然后判断本地存储的 OsKernalFile 文件是否被篡改，如果被篡改，则将 OsKernalFile 文件里的内容清空后向服务器发送请求，要求服务器发送该版本的安全内核文件；如果没有被篡改，再查看本机中是否存储该操作系统的安全内核 Hash 值以及该 Hash 值是否失

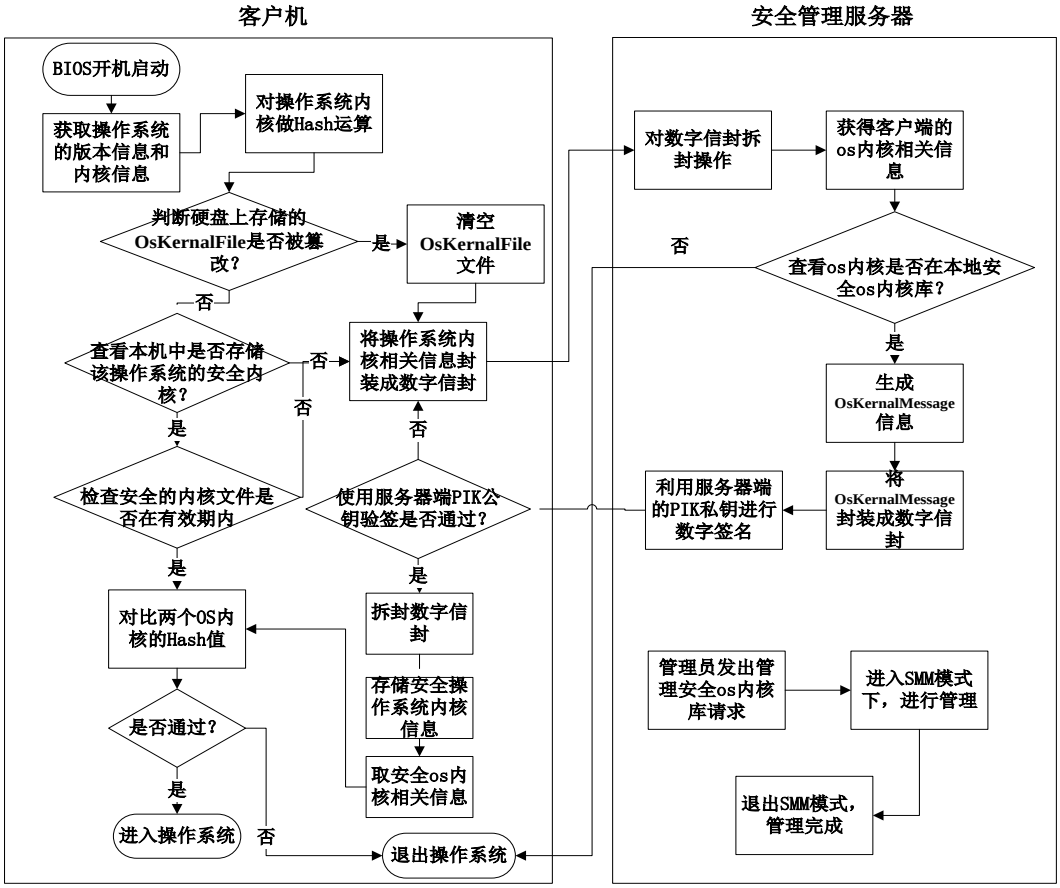

Figure 1. System flow diagram
图 1. 系统流程图

效，如果存在该操作系统的安全内核 Hash 值并且该 Hash 值在有效期内，则取出该安全内核 Hash 值，与开机启动时获取的操作系统内核 Hash 值进行对比，对比通过，进入到操作系统，对比不通过，直接关机；如果不存在该操作系统的安全内核 Hash 值或者该 Hash 值已经失效，则向服务器发送请求，要求服务器发送该版本的安全内核文件。客户机向服务器发送的内容主要包括操作系统的内核 Hash 值和版本信息，安全管理服务器通过通讯模块接收请求，对客户机进行身份的认证，然后得到操作系统内核镜像文件 Hash 值以及版本信息，进一步跟 DB 里存放的安全的操作系统内核镜像 Hash 值进行对比，对比通过之后，将该版本的操作系统内核相关的信息(内核 Hash 值，版本信息和有效期)发送给客户机，客户机接收到该信息之后，将该条信息更新到 OsKernalFile 文件，将整个 OsKernalFile 文件的 Hash 值更新到 TCM 上。

OsKernalFile 文件存储的是客户端装的所有的操作系统内核相关的信息，每一条记录是一个操作系统相关的信息，包括内核的有效期，内核的版本信息和内核的 Hash 值。每次开机，要先对整个文件做 Hash 运算，再跟 TCM 中存储的整个文件的 Hash 做对比，对比通过，说明 OsKernalFile 文件没有被篡改，对比不通过，首先清空该文件，然后再请求服务器重新发送安全内核的 Hash 值。

本系统使用 SM3 杂凑算法[10]产生 Hash 值，SM3 算法是国家密码管理局编制的商用算法，适用于商用密码应用中的数字签名和验证、消息认证码的生成与验证以及随机数的生成，可满足多种密码应用的安全需求。SM3 密码杂凑算法的设计主要遵循的原则有 3 点，第一点是能够有效抵抗比特追踪法及其他分析方法；第二点是软硬件实现需求合理；第三点是在保障安全性的前提下综合性能指标与 SHA-256

同等条件下相当[11]。

安全管理服务器中的 DB 是专门用于存储所有安全的操作系统内核镜像文件的相关信息，包括系统的版本和内核的 Hash 值。在接收验证请求时，首先会对客户端发过来的操作系统内核镜像进行对比，如果在 DB 里能够找到该镜像文件，说明该镜像文件是安全的，将该安全的内核镜像文件发送给客户机，所以要想保证整个过程的一个安全性，首先要确保存放在安全管理服务器上的 DB 的安全性，如果用户攻击了 DB，修改了部分操作系统内核镜像，那么在运行验证内核安全性模块的时候就不会通过，整个过程将会终止，针对这个问题，本系统采取的措施是在硬盘上存储加密的 DB，秘钥存放到 TCM 上，每次修改该 DB 时，都要通过 SMI 中断进入 SMM 模式，在 BIOS 层修改。

整个系统的框架图如图 2 所示，客户机主要包括发送请求模块，安全内核管理模块，信息处理和度量模块，安全管理服务器包括通讯模块，内核安全性验证模块，数据管理模块，发送安全内核相关信息模块。客户机是在操作系统启动之前的 UEFI BIOS 启动的时候进行的一系列的操作，安全管理服务器是在操作系统起来之后的应用层进行的，在 UEFI BIOS 启动的 DXE 阶段，就已经能够进行网络传输，所以向安全管理服务器发送请求模块和安全管理服务器向客户机发送安全内核模块都具有可行性。

3. 基于UEFI固件的操作系统完整性度量的模块设计

3.1. 客户端模块的设计

客户机从本地获取安全的系统内核 Hash 值，通过跟自己机器中的内核 Hash 值对比，决定是否开关机，在本地没有内核 Hash 值时需要向服务器发出请求，要求服务器将安全的内核 Hash 值存储到本机中。如图 3 所示，客户机包括四个模块，分别是请求发送模块，安全内核文件管理模块，信息处理模块，度量模块，其中安全内核管理模块又包括文件的存储管理，失效检测和读取管理 3 个模块。

1) 请求发送模块

该模块主要是用来向服务器端发送请求，请求服务器端将某种版本的操作系统内核文件传输给自己。本地存储的 OsKernalFile 文件被篡改，或不存在该内核 Hash 值或存在但是不在有效期内时调用该模块。

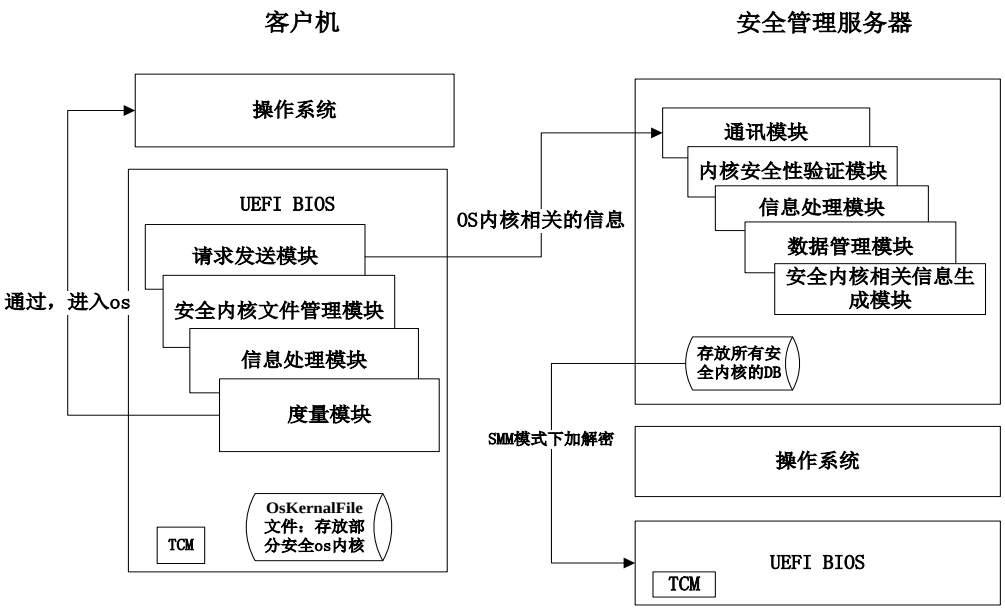

Figure 2. The overall system architecture diagram
图 2. 系统的总体框架图

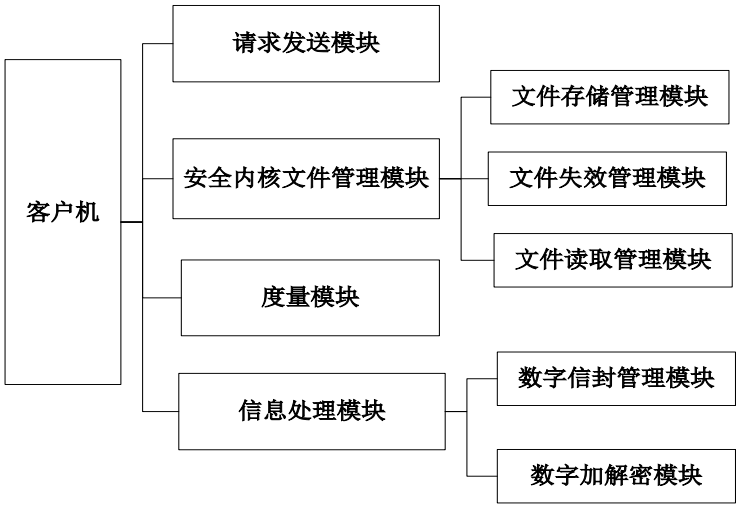

Figure 3. The client organization chart
图 3. 客户端的组织结构图

2) 安全内核管理模块

文件存储管理模块：主要用来存储安全管理服务器端发送过来的安全操作系统内核相关信息的模块。在存储该模块的时候，首先将该条消息存储到硬盘上，然后对整个文件进行杂凑运算得到新的文件 Hash 值，将整个文件 Hash 值更新到 TCM 中。

文件失效检测模块：该模块主要实现检测本机中已存在的安全操作系统内核 Hash 值是否失效，如果失效，调用请求发送模块。主要是通过有效期来判断，首先获得当前的日期，再跟服务器端发来的文件的有效期进行对比。

文件读取管理模块：该模块是用来实现对安全操作系统内核 Hash 值的读取，但是在读取信息之前，需要先确定该 Hash 值是不是被篡改过，需要将存储到本机硬盘上的安全操作系统内核 Hash 值文件在进行杂凑运算，将生成的整个文件的 Hash 值与 TCM 中已存的文件 Hash 值进行对比，对比通过之后，在对硬盘上的安全操作系统内核 Hash 值进行读取操作。

3) 信息处理模块

数据加解密模块：该模块主要实现的是 TCM 相关的一些算法，包括 SM3 杂凑算法，平台身份秘钥 PIK (本质上是一对 SM2 秘钥)。

数字信封解封管理模块：该模块主要实现的功能主要包括两个：一个是对客户端向安全管理服务器端发送信息的封装操作，另一个是对服务器端向客户端发送数据的解封操作。

4) 度量模块

在 BIOS 启动的时候，首先获取 os 内核镜像文件，使用 SM3 杂凑算法产生 Hash 值，调用文件读取管理模块读取文件中的安全 os 内核 Hash 值，将这两个 Hash 值对比，从而决定操作系统的启动与关闭。

3.2. 安全服务器端的设计

服务器端接收到请求之后，将客户端发送过来的内核 Hash 值与本地安全内核 DB 中的同一版本的 Hash 值进行对比，通过之后给客户端发送安全的操作系统内核 Hash 值相关信息。如图 4 所示，服务器端主要包括通讯模块，内核安全性验证模块，信息处理模块，数据管理模块，安全内核相关信息生成模块 5 个模块，其中数据管理模块又包括数据增加模块，数据查找模块和数据删除模块。

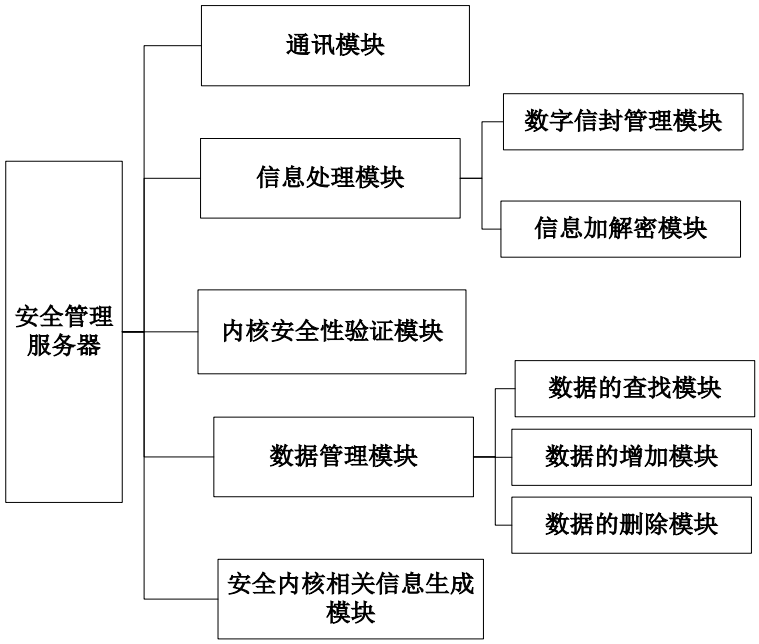

Figure 4. The server module diagram
图 4. 服务器端模块图

1) 通讯模块

主要利用 UEFI 中的网络协议相关的知识实现接收验证请求，以及与客户端的通讯功能。

2) 内核安全性验证模块

该模块主要实现的是将客户端发送的操作系统内核 Hash 值与本地存储的同一版本 os 内核 Hash 值做对比，如果两个 Hash 值一致，说明该客户端发送的内核信息是安全的，如果不一致，则返回一个错误的信息。由于在安全服务器端存放的是已经加密的所有安全操作系统内核 Hash 值文件，所以需要先对其解密。

3) 数据管理模块

该模块是供安全服务器管理员使用的模块，主要实现的是对服务器中的安全操作系统内核 Hash 值文件的管理，包括数据的查找，增加和删除。在服务器端存放的是使用 SM2 公钥对整个文件加密过的，而秘钥存放在 TCM 中。

数据查找模块：该模块主要实现的是内核 Hash 值的获取，首先利用 SMI 中断进入到 SMM 模式，然后在 SMM 模式下获取 SM2 私钥信息，在 BIOS 下对文件进行解密，退出 SMM 模式，找到想要的内核 Hash 值，然后进入到 SMM 模式，在 BIOS 层对文件使用 SM2 公钥加密，最后退出 SMM 模式[9]。

数据增加模块：该模块主要实现的是管理员在 SMM 模式下增加一条或者多条安全的操作系统内核 Hash 值记录，同样需要对安全的操作系统内核 Hash 值整个文件在 UEFI BIOS 层进行解密和加密。

数据删除模块：该模块实现的过程与上述两个模块的实现过程类似，只不过是在对文件的操作不同。在 UEFI BIOS 层进行解密，删除一条信息之后，在对整个存放内核信息的文件进行加密。

4) 安全内核相关信息生成模块

该模块是在跟客户机发送的内核信息比较通过之后调用的模块，主要是用来生成安全内核文件的一些信息，在将这些信息发送到客户机端，生成的安全操作系统内核信息(OsKernalMessage)的格式如图 5 所示，设置分隔符的作用主要是为了便于程序的读取。

内核有效期	分隔符	内核版本信息	分隔符	内核的 hash 值

Figure 5. File format diagram from the server
图 5. 服务器端发送文件格式图

5) 信息处理模块

数字信封封装模块：该模块主要实现的是对客户端发过来的数字信封进行拆封以及将要发送给客户机的信息进行封装。

信息加解密模块：同客户端的信息加解密一样，调用 TCM 相关的算法完成对数据的加解密操作。

4. 基于UEFI固件的操作系统完整性度量流程设计及算法实现

4.1. 客户端存储度量初始值的流程及算法实现

Client：

1. 首先获得操作系统内核信息 OsKernal 和版本信息 OsVersion；

2. 对操作系统内核信息做 SM3 杂凑运算：OsKernalHash = SM3 (OsKernal)；

3. 对本地存储的 OsKernalFile 文件做 hash 运算得到 OsKernalFileHash，跟 TCM 中存储的 OsKernalFileHashTCM 值对比看是否一致，如果不一致，或者一致，但是本地没有存储该操作系统相关的信息，或者存储但是已经失效，拼接系统内核信息和版本信息：OsKernalMsg = OsVersion|OsKernal；

4. 将操作系统内核相关的信息封装成数字信封，封装数字信封的过程如下：

1) 产生一个对称的秘钥，用对称的秘钥对 OsKernalMsg 信息进行加密，利用 TCM 中的随机数生成器生成一个 128 位的数据，将这个数据作为 SMS4 的对称秘钥，利用 SMS4 对称算法对 OsKernalMessage 加密。

encOsKernalMessage = SMS4_Encrypt (SMS4Key，OsKernalMsg)；

2) 利用服务器的 PIK 公钥 Server_PIK_public 对 SMS4Key 对称秘钥进行加密，得到加密的对称秘钥 EncryptSMS4Key，EncryptSMS4Key = SM2_Encrypt (Server_PIK_public，SMS4Key)；

3) 将加密的操作系统内核信息 encOsKernalMsg 和加密秘钥 EncryptSMS4Key 封装成数字信封。

DigitalEnvelop = encOsKernalMsg|EncryptSMS4Key。

5. 将封装好的 DigitalEnvelop 发送给安全管理服务器。

Server。

6. 安全管理服务器接收到 DigitalEnvelop 信息之后，对 DigitalEnvelop 进行拆封运算，具体的流程如下：

1) 拆封数据信封，分别得到 encOsKernalMsg 信息和 EncryptSMS4Key 信息。

2) 使用自己的 PIK 私钥 Server_PIK_private 解密，得到对称秘钥 SMS4Key。

SMS4Key = SM2_Decode (Server_PIK_private，EncryptSMS4Key)。

3) 使用对称秘钥 SMS4Key 解密加密的操作系统内核信息 encOsKernalMsg 得到 OsKernalMsg。

OsKernalMsg = SMS4_Decoode (SMS4Key,encOsKernalMsg)。

7. 查看 os 内核信息 OsKernalMsg 是否在本地库中，String SerchMessage (File f，String str)函数实现的是查找文件中是否存在 s 功能，查找成功返回数据库中的这条记录，并进行下一步信息的封装操作；查找失败返回 NOT_EXIT,并将该消息发送给客户端，客户端收到该消息之后，执行关机命令。

8. 将整条 os 内核记录与有效期进行拼接，得到 OsKernalMessage。

OsKernalMessage = indate|OsVersion|OsKernalHash。

9. 对 OsKernalMessage 进行封装，封装的过程如下，其中 Client_PIK_public 是客户端 PIK 公钥，SMS4Key 是一个 128 位的随机数：

encOsKernalMessage = SMS4_Encrypt (SMS4Key，OsKernalMessage)；

EncryptSMS4Key = SM2_Encrypt(Client_PIK_public, SMS4Key)；

DigitalEnvelop = OsKernalMessage|EncryptSMS4Key。

10. 使用服务器端的 PIK 私钥数字签名，并将签名信息发送给客户端。

SignatureDigitalEnvelop = SM2_Signature (Server_PIK_private，DigitalEnvelop)；

Client。

11. 客户端用服务器端的 PIK 公钥验证签名，对安全管理服务器进行身份认证。

DigitalEnvelop = SM2_VerifySignature (Server_PIK_public，SignatureDigitalEnvelop)。

12. 拆封数字信封，得到 OsKernalMessage 信息，并将该信息存储到硬盘上。

SMS4Key = SM2_Decode (Client_PIK_private，EncryptSMS4Key)；

OsKernalMessage = SMS4_Decoode (SMS4Key，encOsKernalMessage)。

13. 将存储 OsKernalMessage 的整个文件 OsKernalFile 做 hash 运算，得到文件 Hash 值，在将该信息更新到 TCM 中。

OsKernalFileHash = SM3 (OsKernalMessage)。

4.2. 客户机完整性度量的流程及算法实现

1. 客户机开机启动获取内核相关信息并对内核信息进行 hash 运算，得到内核的 hash 值。

OsKernalHash = SM3 (OsKernal)。

2. 对本地存储的 OsKernalFile 文件做 hash 运算得到 OsKernalFileHash，跟 TCM 中存储的 OsKernalFileHashTCM 值是否一致，如果不一致，则清空 OsKernalFile 文件里的内容，然后调用客户端存储度量初始值模块，具体的步骤在上一小节已详细介绍；如果一致，则说明硬盘上存储的没有被篡改。

OsKernalFileHash = SM3 (OsKernalFile)。

3. 在本地 OsKernalFile 文件中取出该操作系统的 OsKernalHash’，将 OsKernalHash 信息与 OsKernalHash’做对比，如果一致，说明内核没有篡改，进入到操作系统；如果不一致，说明内核被篡改，执行关机命令。

5. 系统安全性的核心问题

本系统是在 UEFI BIOS 启动的时候根据本地存放的安全内核的文件，对操作系统内核文件进行完整性度量，度量通过开机启动，要想保证整个过程的安全性，需要做到以下 3 点：

1) 客户端 OsKernalFile 文件的安全存储问题。OsKernalFile 文件中存储的是本计算机中装的所有操作系统内核 Hash 相关的信息，是度量的初始值，如果 OsKernalFile 文件被篡改，度量会失去意义。为了保证该文件的安全性，需要将整个的 OsKernalFile 文件的 Hash 值存储到 TCM 中。在每次读取该文件时，都需要将 OsKernalFile 文件的 Hash 值与 TCM 中的 Hash 值做对比，对比通过，说明存储到该硬盘上的 OsKernalFile 文件是安全的。

2) 服务器端 DB 的安全存储问题。在安全管理服务器端存放着所有安全可信的操作系统内核镜像的文件，每次都要判断客户机发过来的操作系统内核文件是否在该 DB 中，如果有，说明该操作系统内核镜像文件是安全的。如何保障 DB 的安全性是一件极其重要的工作，该系统采用的是硬件防护，将存储

和数据加解密进行有效的隔离，在操作系统上存放的是已加密的文件，在加密和解密的时候在可信的UEFI BIOS 完成。

3) 传输数据的安全性。数据在网络中传输，为了保证数据的安全性，一般采取的是对需要传输的数据进行加密，常见的一些传输加密技术包括数字信封技术，SSL 协议等技术，由于客户机的固件只是在开机启动的时候工作，而 SSL 在传输数据的时候需要三次握手才能够确认其身份，所以在该系统中 SSL 协议不适用，所以本系统采用了数字信封的方式来保证数据传输的安全性。

6. 实验及结果

该系统的安全性在上一章节已经论述，下面我们将对整个系统的功能进行验证：

1) 在一台计算机中装一个 win8 的操作系统，调用该系统，能够正确的对操作系统的完整性度量，计算机正常开机。

2) 在重新开机启动该计算机，也能够实现对操作系统的完整性度量，计算机正常开机。

3) 修改操作系统内核文件，在开机时进行度量，该操作系统的内核摘要跟本地存储的内核度量初始值不一致，度量不通过，计算机不会开机。

4) 修改客户机本地硬盘中存储的内核度量初始值，在开机启动时度量，对 OsKernalFile 文件做 SM3 杂凑，跟 TCM 中存储的 OsKernalFile 文件 hash 不一致，会请求服务器重新发送该操作系统内核信息。

5) 伪装成服务器的身份，发送给客户机不安全的内核信息，客户机收到该消息后，对服务器进行身份认证不通过，会再次请求服务器发送安全内核信息。

7. 总结

OS 的安全性是计算机安全的核心，而内核的安全性是操作系统安全的核心，本文提出了一种基于UEFI BIOS 的操作系统完整性度量机制，实现了在 UEFI BIOS 的启动过程中对操作系统内核镜像文件的完整性度量，保证了操作系统内核的安全性。在实现的整个过程中最主要的是要保证操作系统度量初始值的完整性和安全性，这是该度量机制的难点也是关键点，如果不能够保证度量初始值的正确性，该系统的度量将不再有意义。

该机制会对电脑的开机速度有较大的影响，在后续的工作中，会对该问题进行改进。本机制只是实现对操作系统完整性的度量，但是不能够保证电脑安装操作系统时操作系统的安全性，下一步可以增加操作系统的远程安装功能，进一步保证电脑中操作系统的安全性。

参考文献 (References)

[1] 胡浩, 张敏, 冯登国. 基于信息流的可信操作系统度量架构[J]. 中国科学院大学学报, 2009, 26(4): 522-529.

[2] Smith, S.W. (2004) Outbound Authentication for Programmable Secure Coprocessors. *International Journal of Information Security*, **3**, 28-41. https://doi.org/10.1007/s10207-004-0033-0

[3] Trusted, B.G. (2010) Computing Group. Trusted Platform Module (TPM) Specifications.

[4] Parno, B., Mccune, J.M. and Perrig, A. (2010) Bootstrapping Trust in Commodity Computers. Security and Privacy. IEEE,:414-429.

[5] 国家密码管理局. 可信计算密码支撑平台功能与接口规范[S]. 国家密码管理局, 2007.

[6] Sailer, R., Zhang, X., Jaeger, T., *et al*. (2004) Design and Implementation of a TCG-Based Integrity Measurement Architecture. *Conference on USENIX Security Symposium*, San Diego, 9-13 August 2004, 16.

[7] 戴正华. UEFI 原理与编程[M]. 北京: 机械工业出版社, 2015.

[8] The Unified EFI Forum (2011) Unified Extensible Firmware Interface Specification Version 2.3.1. http://www.uefi.org

[9] 周艺华, 王伟, 王冠, 等. 基于固件的远程身份认证[J]. 信息安全与技术, 2016, 7(3): 35-39.

[10] 国家密码管理局. SM3 密码杂凑算法(SM3 Cryptographic Hash Algorithm)[M]. 国家密码管理局, 2010. http://www.oscca.gov.cn/UpFile/20101222141857786.pdf

[11] 王小云, 于红波. SM3 密码杂凑算法[J]. 信息安全研究, 2016, 2(11): 983-994.