文章编号:1001 - 9081(2010)05 - 1243 - 03

可信平台模块数据保护能力测试与分析

刘安战, 韩玉民
(中原工学院 软件学院, 郑州 450007)
(liuanzhan@126.com)

摘　要:介绍了关于可信计算的概念,并特别针对可信计算平台联盟的可信计算理念,对可信平台进行了阐述。对可信平台模块(TPM)提供的独有特色的数据保护方式 Binding 和 Sealing 进行了测试,通过分析得出基本结论:可信平台在数据保护方面的能力明显不足,可信计算平台模块的计算能力有待于进一步提高。

关键词:可信计算;可信平台模块;数据绑定;数据封印

中图分类号: TP309　　**文献标志码**:A

Test and analysis of trusted platform module data protection

LIU An-zhan, HAN Yu-min
(*Software College, Zhongyuan University of Technology, Zhengzhou Henan 450007, China*)

Abstract: The concept of trusted computing was introduced. In particular, concerning the trusted computing principle of trusted computing platform alliance, the Trusted Platform Module (TPM) was expounded. The unique characteristics of data protection, Binding and Sealing, were tested. The test results show that the trusted platform's data protection capacity is obviously insufficient and the computing power of the trusted computing platform needs to be further improved.

Key words: trusted computing; Trusted Platform Module (TPM); data binding; data sealing

0 引言

可信计算技术最早是从容错计算发展起来的,“容错”确切些说应该是“容许故障”,以 1971 年召开第一届国际容错计算会议(FTCS-1)为起点。1975 商业化的容错机推向市场。到 20 世纪 90 年代,提出了软件容错的问题,进而发展到网络容错。1995 年在 FTCS-15 上,Avizienis 教授等人提出了“可信计算”(Dependable Computing)的概念。文献[1]对可信计算的历史作了比较详细的阐述。

1999 年 10 月,Intel、微软、IBM、HP 和 Compaq 共同发起成立了可信计算平台联盟(Trusted Computing Platform Alliance, TCPA)[2],并提出了“可信计算”(Trusted Computing)的概念。2003 年 4 月,TCPA 重新改组,更名为可信计算联盟(Trusted Computing Group,TCG)。2003 年 10 月,TCG 推出了 TCG 1.2 技术规范。目前已推出 TCG 规范 1.4 版本[3]。

除了 TCG 倡导的可信计算外,对于可信计算还有许多不同理解。文献[4]从计算组件可预测性方面定义可信计算;文献[5]从系统服务的可信赖的可证明性方面定义可信;文献[6]从系统的可靠性、可用性、信息和行为的安全性方面定义可信。TCG 倡导的可信计算得到了更多研究人员和商家的推崇。

TCG 定义的可信硬件平台是可信平台模块(Trusted Platform Module, TPM)。TPM 是一个可信硬件芯片,它嵌入在主板上,包含了 TPM 的可信 PC 平台图。TPM 由运算部件、存储器、I/O、密码运算处理器、随机数产生器和寄存器等部件组成,完成可信度量的存储、可信度量的报告、密钥产生、加密与签名,以及数据安全存储等功能。TPM 是可信计算机的信任根。

TCG 定义的可信软件栈(TCG Software Stack, TSS)是 TPM 平台上的支撑软件,它的主要作用是为其他软件提供方便和统一使用 TPM 的接口。TSS 采用分层的设计结构,可分为 TPM 驱动层、驱动库、核心服务层和服务提供层。TSPI (TCG Service Provider Interface)是 TPM 提供给应用的最高层的接口,使应用程序可以使用 TPM 提供的可信计算功能。本文测试正是通过 TSPI 提供的接口进行的。目前关于可信软件栈的实现主要有 IBM 的 Trousers 项目[7]、Daonity 项目组的 Daonity-TSS[8] 和 IAIK jTSS 项目[9]。

1 Binding 和 Sealing 性能测试与分析

1.1 测试环境

本文测试硬件采用 Thinkpad R60 笔记本电脑,内嵌 TPM 芯片是 Atmel 厂商生产的符合 TCG1.2 规范的可信芯片,内存 1 GB;操作系统 ubuntu 7.04;TSS 采用的是 Trousers 0.2.9。

1.2 Binding 性能测试与分析

Binding[3] 是可信计算技术提供的传统的公钥加密方法,但是 TPM 的密钥管理机制把密钥分为可以移植的密钥和不可以移植的密钥,当加密的公钥对应的私钥不可以移植时,被公钥加密的数据就只能在创建密钥的 TPM 平台上解密,实现数据与硬件平台的绑定。

测试发现,对于数据绑定操作,所用的 TPM 平台只支持 512 位、1 024 位和 2 048 位的密码,而在 TCG 的规范中,密码还可以是 4 096 位、8 192 位和 16 384 位。

在对较大数据块进行 Binding 操作时,必须把其分成较小的数据块分别进行 Binding 操作,然后再重新组装成较大的数据块以便保存到外存设备中。

对于用 512 位密码进行 Binding 操作,每次操作最多只能加密 17 B 的数据块,通过 Tspi_Data_Bind( ) 函数转化后,

收稿日期:2009 - 11 - 30;**修回日期**:2010 - 01 - 22。
作者简介:刘安战(1980 -),男,河南新乡人,硕士,主要研究方向:可信计算、计算机安全、网络安全;　韩玉民(1963 -),男,河南鹤壁人,高级工程师,主要研究方向:计算机网络应用、网络安全、多媒体技术。

Binding 成 256 B 的数据块；对于用 1024 位密码进行 Binding 操作，每次操作最多只能加密 81 B 的数据块，通过 Tspi_Data_Bind( ) 函数转化成 256 B 的数据块；对于用 2048 位密码进行 Binding 操作，每次操作最多只能加密 209 B 的数据块，通过 Tspi_Data_Bind( ) 函数转化成 256 B 的数据块。对数据块较大的数据只能通过把大的数据块分解成小的数据块进行 Binding 操作。本文测试是通过把文件进行分块，对每个数据块进行 Binding 操作，然后把加密后的数据重新组装成新的文件。每个分块的大小采用对应的密钥位数所能进行 Binding 操作的最大字节数，如在采用 512 位密钥进行 Binding 操作时，对文件分块采用每 17 B 一块。表 1 ~ 3 和图 1 给出了测试结果。

表 1　采用 512 位密钥进行 Binding 的执行时间　s

操作	操作的文件大小/KB										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Binding1	0.984	0.995	0.998	0.995	0.997	0.985	0.997	0.989	1.001	1.005	1.004
Binding2	0.979	0.993	0.983	0.986	0.988	1.008	0.989	1.002	0.990	1.001	1.012
Binding3	0.992	0.982	0.994	0.988	0.997	0.998	1.001	0.999	1.001	1.010	1.006
平均时间	0.985	0.990	0.992	0.990	0.994	0.997	0.996	0.997	0.997	1.005	1.007

表 2　采用 1024 位密钥进行 Binding 的执行时间　s

操作	操作的文件大小/KB										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Binding1	0.991	0.993	1.000	0.992	1.001	0.996	0.996	1.001	0.997	1.000	1.006
Binding2	0.990	0.994	0.996	1.001	0.993	0.988	0.989	0.998	1.002	1.003	0.998
Binding3	0.993	0.993	0.992	0.996	0.985	1.000	1.000	0.996	0.997	0.997	1.004
平均时间	0.991	0.993	0.996	0.996	0.993	0.995	0.995	0.998	0.999	1.000	1.003

表 3　采用 2048 位密钥进行 Binding 的执行时间　s

操作	操作的文件大小/KB										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Binding1	1.033	1.043	1.052	1.048	1.048	1.056	1.063	1.061	1.059	1.062	1.058
Binding2	1.038	1.041	1.044	1.053	1.058	1.061	1.054	1.059	1.058	1.060	1.064
Binding3	1.031	1.044	1.040	1.056	1.051	1.058	1.059	1.055	1.060	1.058	1.068
平均时间	1.034	1.043	1.045	1.052	1.052	1.058	1.059	1.058	1.059	1.060	1.063

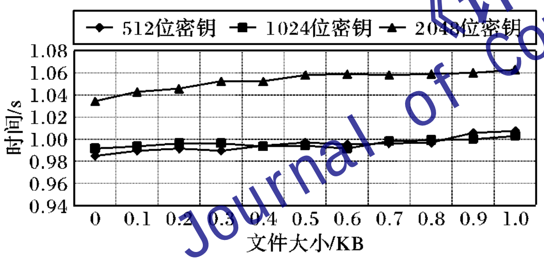

图 1　不同位数密钥的 Binding 执行时间对比

为便于分析，这里说明几个定义。

定义 1　数据膨胀比是指操作后的数据大小与操作前的数据大小比值。膨胀比越小，则操作造成的数据增长也越小，操作造成的存储空间增长也越小，说明性能越好。这里用 E 表示膨胀比，$E(i)$ 表示用 i 位密钥操作时的数据膨胀比。

定义 2　操作字节时间是指每多对一个字节的数据进行操作时所用的时间，平均操作字节时间是操作字节时间平均值，操作字节时间的倒数称为操作字节速率。这里用 TB 表示操作字节时间，$TB(i)$ 表示用 i 位密钥操作时的操作字节时间；VB 表示操作字节速率，$VB(i)$ 表示用 i 位密钥操作时的操作字节速率。

定义 3　操作次时间是指操作字节时间与一次操作时操作的字节数的乘积。在可信数据保护的操作中，对大的数据块是分成小的数据块后，依次进行操作的。平均操作次时间就是操作次时间的平均值，操作次时间的倒数称为操作次速率。这里用 TN 表示操作次时间，$TN(i)$ 表示用 i 位密钥操作时的操作次时间；VN 表示操作次速率，$VN(i)$ 表示用 i 位密钥操作时的操作次速率。

根据定义和测试数据，可得最小数据膨胀比 $E(512) = 256/17 = 15.059$，$E(1024) = 256/81 = 3.160$，$E(2048) = 256/209 = 1.223$。对图 1 中的折线进行线性拟合得到三条拟合后的线性表达式。

512 位密钥的表达式为：

$$T = 0.0019(FS/1024) + 0.9842$$

1024 位密钥的表达式为：

$$T = 0.0009(FS/1024) + 0.9909$$

2048 位密钥的表达式为：

$$T = 0.0025(FS/1024) + 1.0381$$

其中 T 为时间，单位为 s，FS 为文件大小，单位为 B。所以有：

$$TB(512) = 0.0019/1024 = 0.00000185546875$$

$$TN(512) = TB(512) \times 17 = 0.00003154296875$$

$$TB(1024) = 0.0009/1024 = 0.00000087890625$$

$$TN(1024) = TB(1024) \times 81 = 0.00007119140625$$

$$TB(2048) = 0.0025/1024 = 0.00000244140625$$

$$TN(2048) = TB(2048) \times 209 = 0.00051025390625$$

其中 TB、TN 的单位为 s。对照表如表 4 所示。由表 4 可知，2048 位密钥 Binding 操作造成的膨胀比 E 最小，1024 位密钥 Binding 操作对应的 TB 最小，512 位密钥 Binding 操作对应的 TN 最小。

表 4　不同位数密钥的 Binding 操作效率对比

密钥位数	E	TB/μs	TN/μs
512	15.059	1.85546875	31.54296875
1024	3.160	0.87890625	71.19140625
2048	1.223	2.44140625	510.25390625

采用和 Binding 操作分析相同的方法，对 Unbinding 操作进行分析，图 2 是文件大小从 0 ~ 1 KB(1024 B) 的测试结果，

这里的文件大小是在 Binding 前的明文文件的大小。

不同位数密钥的 Unbinding 操作效率对比如表 5 所示。

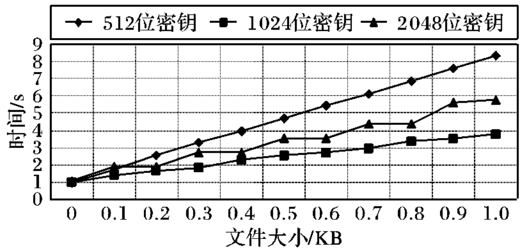

图 2　不同位密钥的 Unbinding 操作时间对比

表 5　不同位数密钥的 Unbinding 操作效率对比

密钥位数	E	TB/ms	TN/ms
512	0.066	0.711 035 156 25	12.087 597 656 25
1 024	0.317	0.270 507 812 50	21.911 132 812 50
2 048	0.816	0.445 019 531 25	93.009 082 031 25

通过对 Binding 和 Unbinding 操作的测试分析，1 024 位密钥测试结果的 TB 都是最小的，因此 1 024 位密钥表现出比较好的时间性能，从图 1 和图 2 中都可以看到 1 024 位密钥操作对相同大小的文件在时间上都占有优势。从图 2 可看到，当数据大小到 1 KB 时，进行 Unbinding 操作的时间最快也要 3.799 s，这个时间已经不能适应很多应用，而采用 512 位密钥时，这个时间已经高达 8.311 s，已经超出了互联网的“八秒原则”，因此，这种情况下已经不能在互联网上展开相应的应用了。图 3 给出了采用更大数据块的测试结果。

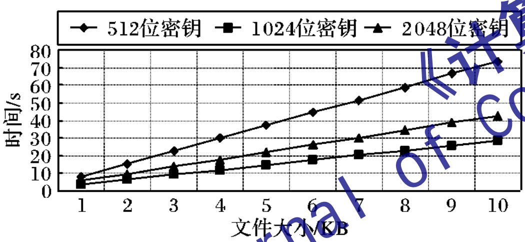

图 3　Unbinding 操作时间与数据块大小的关系

1.3　Sealing 性能测试与分析

Sealing[3] 是可信技术提供的另外一种关于数据加密的方法，它不仅仅可以把数据绑定在特定的可信平台上，而且可以把数据和特定平台的特定状态相关联，TPM 提供了平台状态寄存器（Platform Configuration Register, PCR），该寄存器可以寄存用户环境的当前状态相关信息，状态寄存器的值是基于 SHA 算法通过扩展的方式写入的。Sealing 就是把关键数据和 PCR 值以及加密密钥结合在一起，对数据进行加密，使得加密后的数据依赖于可信平台的状态，使得数据只能在可信的状态下进行解密。

实验表明，针对所用平台数据封印（Sealing）操作，平台只提供了 2 048 位的密码的支持，对于 TCG 规范中提到的其他位数的密码则无法进行 Sealing 操作，Sealing 操作每次最大能加密的数据块是 149 B，加密后转化成 322 B 的数据块，对于大的数据块只能通过分片的形式进行操作。采用 2 048 位密钥进行 Sealing、Unsealing 操作的时间如图 4 所示。

由表 6 可知，对于 Sealing 操作，平均每多增加 1 B，操作时间就会增加 0.16 ms，对应的 Unsealing 操作时间就会增加 24.18 ms。从图 4 可以看到，当数据增加到 1 KB 时，Unsealing 操作已经增加到了 7.730 s。如果采用 Sealing 对1 KB的数据进行保护，Unsealing 操作所用的时间显然是很长的。采用更大的数据块测试结果如图 5 所示，可以看到操作时间与数据块的大小基本成线性关系。

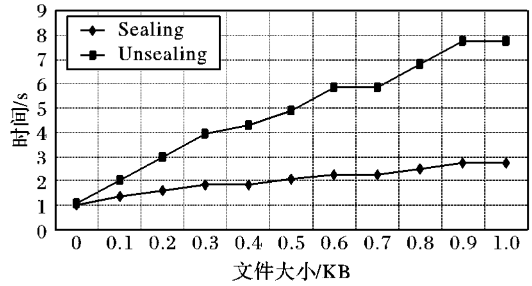

图 4　2 048 位密钥 Sealing 和 Unsealing 的操作时间

表 6　Sealing 和 Unsealing 的操作效率

操作	E	TB/ms	TN/ms
Sealing	2.161	0.162 304 687 5	24.183 398 437 5
Unsealing	0.463	0.645 507 812 5	109.090 820 312 5

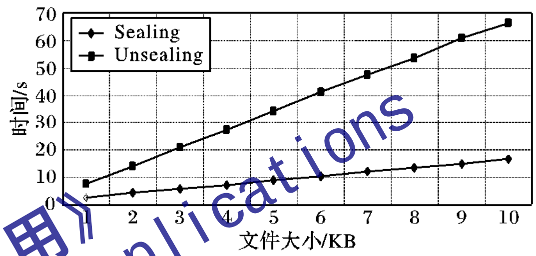

图 5　2 048 位密钥 Sealing 和 Unsealing 操作时间与数据块大小的关系

2　结语

可信计算的理念从其被提出以来，就一直倍受推崇，Binding 和 Sealing 更是可信平台提供的独特的数据保护方式。本文通过对其性能的测试和分析，初步得出结论，目前可信计算提供的数据能力还远远不能满足应用的要求，可信计算技术有待于进一步提高。

参考文献：

[1]　闵应骅. 前进中的可信计算[J]. 中国传媒科技, 2005(9): 50 - 52.

[2]　Trusted computing group [EB/OL]. [2009 - 10 - 18]. http://www.trustedcomputinggroup.org.

[3]　TCG archiecture overview, Version 1.4 [EB/OL]. [2009 - 10 - 18]. http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf

[4]　PERRSON S. Trusted computing platform, the next security solution, HPL-2002-221 [R]. Bristol UK: HP Laboratories, 2002.

[5]　AVIZIENIS A, LAPRIE J C, RANDELL B, *et al*. Basic concepts and taxonomy of dependable and secure computing[J]. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1): 11 - 33.

[6]　张焕国, 罗捷, 金刚, 等. 可信计算研究进展[J]. 武汉大学学报: 理学版, 2006, 52(5): 513 - 518.

[7]　TrouSerS introduction [EB/OL]. [2009 - 09 - 02]. http://trousers.sourceforge.net.

[8]　MAO WENBO, YAN FEI, CHEN CHUNRUN. Daonity: Grid security with behaviour conformity from trusted computing [C]// Proceedings of the first ACM Workshop on Scalable Trusted Computing. New York: ACM Press, 2006: 43 - 46.

[9]　Trusted computing for the Java platform [EB/OL]. [2009 - 09 - 02]. http://trustedjava.sourceforge.net.