密钥安全研究进展

林璟锵　郑昉昱　王跃武
(中国科学院数据与通信保护研究教育中心　北京　100093)
(信息安全国家重点实验室(中国科学院信息工程研究所)　北京　100093)
(中国科学院大学网络空间安全学院　北京　100049)
(linjingqiang@iie.ac.cn)

Advances in Cryptographic Key Protection

Lin Jingqiang, Zheng Fangyu, and Wang Yuewu
(*Data Assurance and Communications Security Research Center*, *Chinese Academy of Sciences*, *Beijing* 100093)
(*State Key Laboratory of Information Security* (*Institute of Information Engineering*, *Chinese Academy of Sciences*), *Beijing* 100093)
(*School of Cyber Security*, *University of Chinese Academy of Sciences*, *Beijing* 100049)

Abstract　In order to achieve the security functionality of cryptographic algorithms, we need to ensure the security of cryptographic keys, i.e., no attacker can access the cryptographic keys. However, there are various attacks access the cryptographic keys on computers that implement cryptographic algorithms and perform cryptographic operations, including system attacks and physical attacks. This paper surveys the attacks that steal cryptographic keys and other sensitive data on computers. We analyze the cryptographic key protections including various solutions based on registers, caches, CPU features, and online central servers and data security techniques on top of protected cryptographic keys, in terms of security, performance and applicability. Finally, we discuss and prospect the research direction of cryptographic key protection in the future.

Key words　cryptographic key protection; applied cryptography; memory attack; system security; physical attack

摘　要　密码算法提供安全功能的前提是密钥数据的安全性,要求攻击者不能获得密钥.然而,在计算机系统中实现密码算法、执行密码计算,面临着各种非授权读取密钥数据的攻击,包括系统攻击和物理攻击.先大致总结了计算机系统中各种窃取密钥以及其他敏感数据的攻击方法;然后重点分析了当前各种典型的密钥安全技术方案,分别包括基于寄存器、基于Cache、基于处理器增强特性、结合中心服务器的解决方案,以及基于密钥安全解决方案的数据安全系统技术,并从安全性、计算性能、适用性等方面对各种方案进行了全面的对比;最后,展望了将来的密钥安全技术研究方向.

关键词　密钥安全;应用密码学;内存攻击;系统安全;物理攻击

中图法分类号　TP309

收稿日期:2018-09-28
基金项目:国家自然科学基金项目(61772518);国家重点研发计划网络空间安全重点专项(2017YFB0802100)

1 从密码算法研究到密钥安全研究

在密码技术发展的早期，密码算法通常都运行在特定的专用密码设备中，例如二战时期的 Enigma 密码机. 对于专用的密码设备，窃取运行期的密钥数据非常困难，攻击者更多通过明密文的数据分析来获得密钥.

随着计算机和网络通信技术的发展，各种密码算法实现为计算机系统上的二进制代码，密钥数据作为进程的数据变量参加密码计算. 此时，从计算机进程、操作系统和 CPU 任务的角度而言，密钥数据与其他的内存数据并没有区别，仅仅是从进程隔离的角度来防范非授权访问. 然而，作为关系到大量数据的机密性、完整性和起源鉴别的关键数据，密钥具有更高的安全要求. 例如，一旦攻击者获得短短的 128 b 对称密钥或者 2 048 b 非对称密钥，就能够危及几 MB、几 GB 甚至几 TB 数据的安全性.

在计算机系统中，对密码计算运行期的密钥数据实施更高强度的安全措施，防范各种恶意攻击行为窃取密钥，才能保证密码技术发挥应有的安全作用；否则，即使密码算法是安全的、安全强度达到 256 b 甚至更高，也是形同虚设.

2 密钥数据面临的各类攻击

本节先简要说明常规密码软件实现中密钥数据的处理，然后列举了各种窃取密钥数据的攻击方法.

2.1 常规密码软件实现

对于常用的密码软件实现，例如公开源代码的 OpenSSL，Crypto++等，其中的密钥数据也是定义为普通的程序变量，密钥并没有得到专门的安全保护，与其他的内存数据相同对待.

现代操作系统通常也提供密码计算服务，例如 Windows 系统的 CSP 和 CNG 服务，在操作系统中对密码计算任务实施保护；但是，这些保护措施是针对完整的计算任务、而不是专门针对其中的密钥数据，保护力度有限.

总体而言，大量的密码软件实现更注重于各种密码算法的计算流程实现，并没有对其中的密钥数据实施专门的安全措施，尤其是运行期的非授权恶意读取.

下面，我们列举各种在计算机系统中、从密码软件实现中窃取密钥数据的攻击. 原则上，由于在常规密码软件实现中密钥数据作为普通的内存数据而存在，这些攻击行为也可用于窃取内存中的其他敏感数据.

2.2 系统攻击

利用计算机系统的软硬件漏洞，有大量的系统攻击可以窃取系统中的敏感数据. 由于操作系统和各种应用软件的复杂性，难以绝对消除其中的安全漏洞，所以相应的攻击行为也会长期存在.

即使在不破坏二进制代码完整性的前提下，攻击者可以利用以下各种类型的安全漏洞来越权访问密钥数据：

1）基于进程隔离漏洞. 例如，操作系统漏洞[1-4]导致用户态的攻击程序可以访问其他进程的内存数据.

2）基于内存数据扩散. 计算机系统的休眠、交换分区、Core Dump 功能会导致内存数据扩散到硬盘，程序崩溃报告也经常包含敏感的内存数据并经公开网络信道发送[5].

3）基于未清零的动态内存数据. 动态内存区域使用后，如果没有清零，该内存区域有可能分配给攻击者程序导致数据泄露[6-7].

4）基于密码软件的自身安全漏洞. 最典型的实例就是影响广泛的 OpenSSL Heartbleed 漏洞，由于软件没有正确处理畸形数据包，攻击者可以获得随机地址的 64KB 数据.

以上的攻击操作中，攻击者不需要篡改二进制代码或者破坏控制流完整性. 此外，攻击者还可以利用恶意代码注入、控制流劫持[8]等破坏二进制代码执行的攻击方法来非法访问密钥数据.

2.3 物理攻击

物理攻击是指攻击过程要求攻击者能够接触被攻击计算机. 最典型的窃取密钥数据的物理攻击是冷启动攻击[9]. 冷启动攻击利用内存芯片的延迟消失效应：断电之后，内存芯片上的数据仍然会维持一段时间，在低温下甚至可以长达数小时. 攻击者获得正在运行的计算机，就可以插入恶意的存储介质（带有启动代码），然后强制让计算机从该存储介质引导启动，加载攻击代码，并立即读

取内存芯片中尚未消失的原有数据.

DMA 攻击是另一大类针对内存数据的物理攻击[10-11]. DMA 特性的初衷是在外设与内存数据之间建立高速的、不需 CPU 参与的传输通道. 利用 DMA 特性,攻击者插入恶意外设,可以绕过操作系统的访问控制,直接读取内存数据. DMA 攻击还可以向受害计算机的内存空间写入恶意代码,然后执行该恶意代码就可以读取寄存器中的敏感数据,将攻击目标范围从内存数据扩展到寄存器数据[12]. DMA 攻击甚至可以利用有漏洞的外设,在没有物理接触的情况下发起:攻击者利用系统攻击远程地控制受害计算机上已有的、有漏洞的外设,然后发起 DMA 攻击[13].

3 密钥安全解决方案

下面,我们列举给出多种现有公开文献的密钥安全解决方案. 不同的密钥安全解决方案,适用于不同的密码算法和不同的计算平台.

3.1 基于寄存器的解决方案

解决冷启动攻击的基本思路就是将密码计算限制在 CPU 内部,不使用内存芯片. TRESOR 方案[14]利用寄存器,结合 Intel CPU 内置的 AES-NI 指令,完成了基于寄存器的 AES 密码软件引擎,并用于全磁盘数据加密. AES 主密钥始终存储在特权的 DEBUG 寄存器上;需要执行 AES 密码计算时,利用 AES-NI 指令和向量寄存器来完成. 在 AES 密码计算过程中,保证主密钥以及轮密钥只出现在寄存器上. Amnesia 方案[15]在 AMD CPU 上也实现了类似的密码软件,利用 MSR 寄存器来存储 AES 主密钥,同样保证在密码计算过程中,主密钥以及轮密钥只出现在寄存器上.

PRIME 方案在 Intel 平台上完成了基于寄存器的 RSA 算法[16]. 相比 AES 算法,RSA 算法需要更大的数据空间,PRIME 方案利用 AVX 向量寄存器来执行 RSA 计算;由于可用的寄存器数量限制,PRIME 方案的执行效率远远低于常规的密码软件实现. RegRSA 方案[17]大幅度地改进了 PRIME 方案的效率:引入更多的寄存器参与 RSA 密码计算,将部分中间计算结果加密后存储到内存中,使得计算效率接近于常规的、无特殊密钥安全措施的软件实现.

3.2 基于 Cache 的解决方案

CPU 内部的 Cache 也可以作为存储空间,完成密码计算,从而实现不使用内存芯片的密钥安全方案,抵抗冷启动攻击. FrozenCache 方案[18]提出,在锁屏和待机时,将全磁盘数据加密的 AES 密钥存储在 Cache 上,避免冷启动攻击.

Copker 方案[19]第 1 次完成了在 Cache 上执行的公钥密码计算. 与 PRIME 方案类似,Copker 方案利用 TRESOR 方案在寄存器上存储 AES 主密钥作为密钥加密密钥 Key-Encryption Key,在需要执行 RSA 密码计算时,将密文状态的 RSA 私钥解密到内存空间,通过配置 Cache 工作模式,使得 RSA 私钥不会被同步到内存芯片上;同时还需要配置其他核的 Cache 工作模式,避免不同核之间的 Cache 数据同步影响. Copker 方案的计算效率接近于常规的、无特殊密钥安全措施的软件实现. 由于公钥密码计算的算法实现不限制特定寄存器,可以使用高级编程语言,实现快捷,容易支持各种公钥密码算法(相比而言,基于寄存器的解决方案需要使用汇编语言来实现公钥密码算法). 而且,由于 Cache 空间几乎无限(L1 数据 Cache 有 32 KB),可以支持各种密码算法以及安全增强实现[20].

Sentry 方案[21]是基于 ARM CPU 的数据安全方案,针对冷启动攻击和 DMA 攻击. 其技术思路也是利用 Cache 来存储数据明文,相应的数据密文存储在内存芯片中,所以用于数据加密的 AES 对称密钥也应该限制在 Cache 上. Sentry 方案利用了 ARM CPU 的 Cache Locking 特性,可以将数据"锁定"在 Cache 上,而不会被同步到内存芯片.

3.3 基于处理器增强特性的解决方案

随着硬件能力的不断提升,有越来越多的增强特性实现在 CPU 上. 利用不同的处理器增强特性,可以实现密钥安全方案.

Mimosa 方案[22]利用 Intel TSX 特性实现了基于硬件事务内存的密钥安全方案. 硬件事务内存技术的提出,其初衷是为了解决多线程程序的数据共享读写;当一个任务处于事务状态时,CPU 硬件会自动地记录该任务所读写的所有数据地址,一旦有其他任务有数据读写冲突(例如,在事务未结束时,其他任务读取该任务更新的数据或者更改该任务读取的数据),该任务就会自动回滚

到事务开始时的状态. 类似于公钥密码计算的其他密钥安全方案,Mimosa 方案也利用特权寄存器上存储 AES 密钥作为密钥加密密钥,没有计算任务时,被保护的 RSA 私钥处于密文状态. Mimosa 方案在事务状态中执行 RSA 密码计算,由于 RSA 私钥以及各种敏感的中间计算变量都是事务执行过程中更新的数据,任何恶意程序的非授权读取(来自于恶意软件或者 DMA 请求)都会导致事务回滚,攻击者只能得到事务开始前的数据. 而且,由于 Intel TSX 特性事务执行的中间状态自动存储在 Cache 上,而不在内存芯片,Mimosa 方案也同时能够抵抗冷启动攻击.

TrustOTP 方案[23]利用 ARM TrustZone 机制安全模式,在内存空间中创建隔离的计算环境,用于密码计算,实现一次性动态口令;同时,还实现了可信显示,输出动态口令计算结果. 由于 ARM TrustZone 机制的隔离能力,运行在普通模式的恶意软件(包括恶意操作系统)不能读取其中的密钥等安全数据. 同样基于 ARM TrustZone 机制的内存访问控制能力,TrustICE 方案[24]实现了运行在普通模式的隔离计算环境,用于密码计算,在提供安全能力的同时不会无限增加安全模式的可执行代码. CaSE 方案[25]扩展了 TrustOTP 方案,进一步利用 ARM CPU 特性将安全模式运行的程序限制在 Cache 上,同时抵抗冷启动攻击和普通模式的软件攻击. TrustOTP 方案、TrustICE 方案和 CaSE 方案也可用于其他具有安全需求的计算任务.

不同于以上的密钥安全方案,PixelVault 方案[26]基于 GPU 实现了安全的密码计算环境,保护其中的密钥数据. PixelVault 方案在 GPU 寄存器上实现密码计算,由于 GPU 的寄存器数量大,可以实现各种密码算法,并且将代码加载到 GPU 指令 Cache 上,利用 GPU 的非抢占执行模式,使得主机 CPU 上的恶意攻击无法从 GPU 寄存器上读取密钥,也不能改变 GPU 上的可执行代码. 虽然文献[27]表明 PixelVault 方案假设的 GPU 特性并不成立,利用 NVIDIA 公司未正式公开的技术可以改变非抢占执行模式的可执行代码,可以利用调试模式从 GPU 寄存器上读取数据,但是 GPU 作为常见的高性能密码计算器件,研究相应的密钥安全方案具有重要意义.

3.4 结合中心服务器的解决方案

利用中心服务器,解决在线移动终端的密钥安全问题,在终端丢失的情况下,可以及时禁用或者销毁密钥. mRSA 方案[28]利用门限密码技术,将 RSA 私钥拆分为 2 份,由 Semi-Trusted 的中心服务器和终端共同掌握:任何一方都持有完整的私钥,每一次签名或者解密计算都需要双方协作完成,而且在协作计算过程中不会泄露任何有关私钥的信息. mRSA 方案可以实现对 RSA 私钥的使用控制:在通常状态下,中心服务器配合终端的密码计算;一旦出现任何安全问题,就可以及时禁用或者销毁密钥. 该技术思想可以应用到其他公钥密码算法,包括 ECDSA 算法[29]、SM2 算法[30]、基于标识的公钥密码算法[31]等.

CleanOS 方案[32]利用中心服务器,采取了不同的解决思路:在移动终端中使用完整的对称密钥来加解密敏感数据,不同的数据对象使用不同的、相互独立的对称密钥来处理;同时,将对称密钥托管到可信的中心服务器,并在移动终端上删除密钥,只有在需要使用数据时,才从中心服务器请求密钥,解密数据;如果数据在一定时限内不使用,就删除相应的对称密钥. 移动设备使用的对称密钥仅在必要的时间内出现和使用. 一旦移动设备丢失,通过中心服务器的密钥禁用就可以防止攻击者获得明文数据.

3.5 基于密钥安全解决方案的系统安全技术

在计算机系统上,一旦密钥安全问题得到解决,就能够在此基础上实现更为全面的系统安全技术. 例如,上文所述的 TRESOR 方案、Amnesia 方案、FrozenCache 方案,在全磁盘数据加密中得到应用;Copker 方案和 Mimosa 方案在 HTTPS 服务中保护服务器的 RSA 私钥.

虚拟化技术为了大量的系统安全问题提供了新的解决思路和技术途径,同样也给密钥安全研究带来了不同的解决方案. 由于全磁盘数据加密的密钥安全方案[14-15]需要更新操作系统内核,难以应用于 Windows 等闭源操作系统,所以 TreVisor 方案[33]结合了 TRESOR 方案和 BitVisor 虚拟化技术[34],在虚拟机监控器中实现了可抵抗冷启动攻击的密钥安全方案,为上层的各种虚拟机提供更安全的磁盘数据服务. 文献[35]结合虚拟化技术,基于 GPU 向虚拟机提供了密钥安全的密码计算

服务. 首先,密钥仅在虚拟机监控器的内存空间中出现,虚拟机不能访问;其次,为了保证 GPU 上运行的可执行代码不会恶意地泄露密钥,由虚拟机监控器来验证其完整性;最后,使用 API Remoting 技术向虚拟机提供 GPU 计算接口,实现 GPU 设备虚拟化.

CCFI 方案[36]在保留寄存器中随机生成并存储 AES 密钥,基于该密钥动态地生成并校验各种控制流数据(如返回地址、函数指针等). 由于 AES 密钥存储在由编译器保留的 XMM 寄存器中,程序中不包含其他使用 XMM 寄存器的代码,攻击者也无法构造能够通过校验的控制流劫持.

基于 TRESOR 方案保护的对称密钥,RamCrypt 方案[37]在 Intel CPU 上实现了 Linux 操作系统的进程内存数据加密;RamCrypt 方案不需要 Intel SGX 特性的支持,不需要用户进程重新编译,但是每一个进程仍然会有少量页的数据处于明文状态. 类似地,CryptMe 方案[38]利用 ARM TrustZone 机制来保护对称密钥,进而在 ARM 设备上实现了 Linux 操作系统的进程内存数据加密.

4 研究总结和展望

不同的密钥安全方案适用于不同的密码算法、应用场景和安全服务. 文献[39]对多种密钥安全方案给出了全面的对比分析和总结.

如前文所述,在专用的密码硬件设备上,如密码芯片、USB Token、HSM(hardware security module)等,密钥安全并不突出;密钥安全方案更多是针对通用计算平台上的软件实现的密码计算. 随着 Intel SGX 特性、AMD SME/SEV 特性等通用 CPU 内置的内存数据加密等安全特性的引入,应用程序中的敏感数据获得了更高的安全性;相应地,在 SGX Enclave 等安全环境中实现的密码软件也会得到更好的密钥安全保证. 然而,新近出现的 Meltdown 和 Spectre 硬件安全漏洞[40-42]表明,攻击者可以通过更多的途径来窃取密钥等敏感数据,相应的攻击也可以突破 Intel SGX 特性的安全保护;AMD SME/SEV 特性也存在不同程度的安全漏洞[43-44]. 目前而言,如何实现抵抗 CPU 硬件安全漏洞的密钥安全方案,仍然是重要的技术挑战,仍然值得我们继续努力.

参 考 文 献

[1] Guninski G. Linux kernel 2.6 fun, Windoze is a joke [EB/OL]. (2005-02-15) [2018-09-20]. http://www.guninski.com/where_do_you_want_billg_to_go_today_3.html

[2] Lafon M, Francoise R, CAN-2005-0400: Information leak in the Linux kernel ext2 implementation [EB/OL]. (2005-03-25) [2018-09-20]. https://seclists.org/bugtraq/2005/Apr/17

[3] National Vulnerability Database. CVE-2014-0069 [EB/OL]. (2014-02-28) [2018-09-20]. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0069

[4] National Vulnerability Database. CVE-2014-4653 [EB/OL]. (2014-03-07) [2018-09-20]. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-4653

[5] Chow J, Pfaff B, Garfinkel T, et al. Understanding data lifetime via whole system simulation [C] //Proc of the 13th USENIX Security Symp. Berkeley, CA: USENIX Association, 2004: 321-336

[6] The MITRE Corporation. CWE-212: Improper cross-boundary removal of sensitive data [EB/OL]. (2018-03-29) [2018-09-20]. https://cwe.mitre.org/data/definitions/212.html

[7] The MITRE Corporation. CWE-226: Sensitive information uncleared before release [EB/OL]. (2018-03-29) [2018-09-20]. https://cwe.mitre.org/data/definitions/226.html

[8] Szekeres L, Payer M, Wei T, et al. Sok: Eternal war in memory [C] //Proc of the 34th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2013: 48-62

[9] Halderman J A, Schoen S D, Heninger N, et al. Lest we remember: Cold-boot attacks on encryption keys [J]. Communications of the ACM, 2009, 52(5): 91-98

[10] Stewin P, Bystrov I. Understanding DMA malware [C] //Proc of Int Conf on Setection of Intrusions and Malware, and Vulnerability Assessment. Berlin: Springer, 2012: 21-41

[11] Becher M, Dornseif M, Klein C N. FireWire: All your memory are belong to us [OL]. 2005 [2018-09-20]. https://cansecwest.com/core05/2005-firewire-cansecwest.pdf

[12] Blass E O, Robertson W. TRESOR-HUNT: Attacking CPU-bound encryption [C] //Proc of the 28th Annual Computer Security Applications Conf. New York: ACM, 2012: 71-78

[13] Li Y, McCune J M, Perrig A. VIPER: Verifying the integrity of peripherals' firmware [C] //Proc of the 18th ACM Conf on Computer and Communications Security. New York: ACM, 2011: 3-16

[14] Müller T, Freiling F C, Dewald A. TRESOR runs encryption securely outside RAM [C] //Proc of USENIX Security Symp. Berkeley, CA: USENIX Association, 2011

[15] Simmons P. Security through Amnesia: A software-based solution to the cold boot attack on disk encryption [C] // Proc of the 27th Annual Computer Security Applications Conf. New York: ACM, 2011: 73-82

[16] Garmany B, Müller T. PRIME: Private RSA infrastructure for memory-less encryption [C] //Proc of the 29th Annual Computer Security Applications Conf. New York: ACM, 2013: 149-158

[17] Zhao Y, Lin J, Pan W, et al. RegRSA: Using registers as buffers to resist memory disclosure attacks [C] //Proc of IFIP Int Information Security and Privacy Conf. Berlin: Springer, 2016: 293-307

[18] Pabel J. FrozenCache: Mitigating cold-boot attacks for full-disk-encryption software [C/OL] //Proc of the 27th Chaos Communication Congress. 2010 [2018-09-20]. https://events.ccc.de/2010/12/28/frozen-cache/

[19] Guan L, Lin J, Luo B, et al. Copker: Computing with private keys without RAM [C] //Proc of the 21st Annual Network and Distributed System Security Symp. Rosten: The Internet Society, 2014: 23-26

[20] Lin J, Guan L, Ma Z, et al. Copker: A cryptographic engine against cold-boot attacks [J]. IEEE Trans on Dependable and Secure Computing, 2018, 15(5): 742-754

[21] Colp P, Zhang J, Gleeson J, et al. Protecting data on smartphones and tablets from memory attacks [C] //Proc of the 20th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2015: 177-189

[22] Guan L, Lin J, Luo B, et al. Protecting private keys against memory disclosure attacks using hardware transactional memory [C] //Proc of IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2015: 3-19

[23] Sun H, Sun K, Wang Y, et al. TrustOTP: Transforming smartphones into secure one-time password tokens [C] // Proc of the 22nd ACM Conf on Computer and Communications Security. News York: ACM, 2015: 976-988

[24] Sun H, Sun K, Wang Y, et al. TrustICE: Hardware-assisted isolated computing environments on mobile devices [C] //Proc of the 45th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks. Piscataway, NJ: IEEE, 2015: 367-378

[25] Zhang N, Sun K, Lou W, et al. CaSE: Cache-assisted secure execution on arm processors [C] //Proc of IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2016: 72-90

[26] Vasiliadis G, Athanasopoulos E, Polychronakis M, et al. PixelVault: Using PGUs for securing cryptographic operations [C] //Proc of the 2014 ACM Conf on Computer and Communications Security. New York: ACM, 2014: 1131-1142

[27] Zhu Z, Kim S, Rozhanski Y, et al. Understanding the security of discrete GPUs [C] //Proc of the General Purpose GPUs. New York: ACM, 2017: 1-11

[28] Boneh D, Ding X, Tsudik G, et al. A method for fast revocation of public key certificates and security capabilities [C] //Proc of USENIX Security Symp. Berkeley, CA: USENIX Association, 2001: 22-22

[29] Lindell Y. Fast secure two-party ECDSA signing [C] // Proc of Annual International Cryptology Conf. Berlin: Springer, 2017: 613-644

[30] 林璟锵, 马原, 荆继武, 等. 适用于云计算的基于 SM2 算法的签名及解密方法和系统: 中国, ZL2014104375995 [P]. 2017-11-03

[31] Libert B, Quisquater J J. Efficient revocation and threshold pairing based cryptosystems [C] //Proc of the 22nd Annual Symp on Principles of Distributed Computing. New York: ACM, 2003: 163-171

[32] Tang Y, Ames P, Bhamidipati S, et al. CleanOS: Limiting mobile data exposure with idle eviction [C] //Proc of the 10th USENIX Symp on Operating Systems Design and Implementation. Berkeley, CA: USENIX Association, 2012: 77-91

[33] Müller T, Taubmann B, Felix C. Freiling. TreVisor - OS-independent software-based full disk encryption secure against main memory attacks [C] //Proc of Int Conf on Applied Cryptography and Network Security. Berlin: Springer, 2012: 66-83

[34] Shinagawa T, Eiraku H, Tanimoto K, et al. BitVisor: A thin hypervisor for enforcing i/o device security [C] //Proc of the 2009 ACM SIGPLAN/SIGOPS Int Conf on Virtual Execution Environments. New York: ACM, 2009: 121-130

[35] Wang Z, Zheng F, Lin J, et al. Utilizing GPU virtualization to protect the private keys of GPU cryptographic computation [C] //Proc of Int Conf on Information and Communications Security. Berlin: Springer, 2018: 142-157

[36] Mashtizadeh A J, Bittau A, Boneh D, et al. CCFI: Cryptographically enforced control flow integrity [C] // Proc of the 22nd ACM Conf on Computer and Communications Security. New York: ACM, 2015: 941-951

[37] Götzfried J, Müller T, Drescher G, et al. RamCrypt: Kernel-based address space encryption for user-mode processes [C] //Procs of the 11th ACM on Asia Conf on Computer and Communications Security. New York: ACM, 2016: 919-924

[38] Cao Chen, Guan Le, Zhang Ning, et al. CryptMe: Data leakage prevention for unmodified programs on ARM devices[G]// LNCS 11050: Proc of Int Symp on Recent Advances in Intrusion Detection. Berlin: Springer, 2018: 380-400

[39] Lin Jingqiang, Luo Bo, Guan Le, et al. Secure computing using registers and caches: The problem, challenges, and solutions[J] IEEE Security & Privacy, 2016, 14(6): 63-70

[40] Lipp M, Schwarz M, Gruss D, et al. Meltdown [EB/OL]. (2018-01-03) [2018-09-20]. https://arxiv.org/abs/1801.01207

[41] Kocher P, Genkin D, Gruss D, et al. Spectre attacks: Exploiting speculative execution [EB/OL]. (2018-01-03) [2018-09-20]. https://arxiv.org/abs/1801.01203

[42] Chen Guoxing, Chen Sanchuan, Xiao Yuan, et al. SgxPectre attacks: Stealing Intel secrets from SGX enclaves via speculative execution [EB/OL]. (2018-02-25) [2018-09-20]. https://arxiv.org/abs/1802.09085

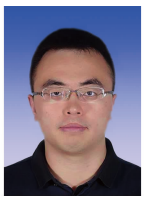
[43] Hetzelt F, Buhren R. Security analysis of encrypted virtual machines [C] //Proc of the 13th ACM SIGPLAN/SIGOPS Int Conf on Virtual Execution Environments. New York: ACM, 2017

[44] Morbitzer M, Huber M, Horsch J, et al. SEVered: Subverting AMD's virtual machine encryption [EB/OL]. (2018-05-24) [2018-09-20]. https://arxiv.org/abs/1805.09604

林璟锵

博士，研究员，主要研究方向为应用密码学、网络与系统安全.

linjingqiang@iie.ac.cn



郑昉昱

博士，助理研究员，主要研究方向为应用密码学、高性能计算和计算机算术.

fyzheng@is.ac.cn

王跃武

博士，研究员，主要研究方向为移动安全、容器安全以及密码应用测评.

wangyuewu@iie.ac.cn