计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(5):65−74 [doi: 10.15888/j.cnki.csa.008550]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

抗量子可信计算安全支撑平台技术①

李　为[1,2], 齐　兵[1,2], 秦　宇[2], 冯　伟[2]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 软件研究所 可信计算与信息保障实验室, 北京 100190)
通信作者: 李　为, E-mail: liwei2018@iscas.ac.cn

摘　要: 随着科技的发展, 量子计算机大规模部署逐渐变为可能, 基于部分计算困难问题的公钥密码算法将被量子算法有效求解. 传统的可信硬件芯片如 TCM/TPM 等由于广泛使用了 RSA、SM3、ECC 等公钥密码体制, 其安全性将受到严重影响; 而绝大部分具有抗量子能力的密码算法并不适配现有 TCM/TPM 芯片有限的计算能力, 因此需要对抗量子可信计算平台进行重新设计. 本文针对可信计算在量子计算模型下面临的安全挑战, 分析总结了抗量子可信计算的研究现状, 改进并提出了抗量子可信计算技术体系, 并结合现有的后量子密码算法协议和可信计算软硬件技术框架, 通过在可信计算平台上移植抗量子密码算法和协议, 实现了基于 TCM 的抗量子可信计算安全支撑平台, 包括可信密码模块本原根设计, TCM 密码库、远程证明、LDAA 等抗量子可信计算扩展功能改进. 最后在可信计算仿真平台上对信任根、软件库、远程证明等抗量子 TCM 模块的功能和性能进行了全面测试, 结果表明平台既具有抵抗量子算法攻击的安全性, 且具有可以接受的应用性能开销.

关键词: 抗量子密码算法; 抗量子可信密码模块; TPM/TCM; 可信计算安全支撑平台; 远程证明; 信息安全

引用格式: 李为,齐兵,秦宇,冯伟.抗量子可信计算安全支撑平台技术.计算机系统应用,2022,31(5):65–74. http://www.c-s-a.org.cn/1003-3254/8550.html

Technology of Quantum-resistant Trusted Computing Security Support Platform

LI Wei[1,2], QI Bing[1,2], QIN Yu[2], FENG Wei[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: With the development of science and technology, the deployment of large-scale quantum computers is becoming possible, and the public-key cryptographic algorithms based on some difficult problems will be solved by quantum algorithms effectively. The security of traditional trusted hardware chips such as TCM/TPM will be seriously affected due to the wide use of public-key cryptosystems such as RSA, SM3, and ECC, and most of the quantum-resistant (QR) cryptographic algorithms cannot be implemented on hardware chips with limited computational resources. Therefore, it is necessary to redesign the QR trusted computing platform. In this study, considering the security challenges faced by trusted computing in quantum computing models, we summarize the present situation of QR trusted computing research and propose a QR trusted computing technology system. Combined with the existing post-quantum cryptographic protocol and trusted computing software and hardware technology framework, we transplant the QR cryptographic algorithms and protocol on the trusted computing platform and implement a prototype system of a QR trusted computing security support platform based on TCM. The work includes the design of the primitive root key and QR extensions such as TCM cipher library, remote attestation, and LDAA. Finally, the results of function and

① 基金项目: 国家重点研发计划 (2020YFE0200600); 国家自然科学基金 (61872343, 61802375); 中国科学院青年创新促进会资助项目
收稿时间: 2021-07-02; 修改时间: 2021-08-17, 2021-09-28, 2021-10-26; 采用时间: 2021-11-07; csa 在线出版时间: 2022-02-25

performance tests on the emulator for the above TCM modules show that the prototype system is resistant to attacks by quantum algorithms, with acceptable application performance overhead.

Key words: quantum-resistant cryptographic algorithm; quantum-resistant trusted cryptographic platform; TPM/TCM; trusted computing security support platform; remote attestation; information security

1 背景和现状

随着量子计算理论的发展, 一些经典模型下的困难问题被发现在量子计算模型下可以被有效求解. 1999 年, Shor 发现了多项式时间内用于解决大整数分解问题和离散对数问题的量子算法[1]. 在大规模量子计算机上使用这类算法可以破坏被广泛应用的 RSA、ECC 等密码算法以及 DH 密钥交换等协议, 而这些都是现今通信协议所依赖的核心功能: 公钥加密, 数字签名和密钥交换. 当量子计算机的规模扩大到一定程度, 现有的公钥密码体制将被完全破坏.

2019 年谷歌在具有 53 个量子比特的处理器上用 200 s 完成了在经典超级计算机上需要执行一万年才能完成的运算, 并声称这是量子计算机对传统计算机的根本性优势[2]. 对于对称密码系统, 与传统计算机上的搜索算法相比, Grover 算法[3] 仅为量子搜索算法提供了 2 倍性能的加速. 对此将对称密钥长度加倍就足以保持原有的安全性. 此外, 已有研究表明, 不可能以指数级的速度加快搜索算法的速度[4], 这表明对称算法和哈希函数应该在量子时代可用.

随着量子计算和量子计算机的深入研究, 作为可信计算安全基础的密码算法和可信硬件信任根将受到严重的安全挑战.

1.1 量子计算机对可信计算的影响

2001 年, 国际可信计算产业联盟 (TCG 前身) 推出了 TPM (trusted platform module, 可信平台模块) 技术标准. 我国于 2007 年 12 月发布了《可信计算密码支撑平台功能与接口规范》[5], 推出了通用的计算机信任根 TCM (trusted cryptography module, 可信密码模块). TPM/TCM 作为通用信任根在计算机产业界得到了广泛的应用, 全球主流的个人电脑、服务器上几乎都配置部署了此类安全芯片.

TCM 是我国在可信计算领域自主研制的安全芯片. TCM 借鉴了国际可信计算技术框架与技术理念, 以国产密码算法为基础, 安全性与计算效率都比 TPM 有较大提高.

TPM2.0 新规范在 2014 年由 TCG 提出[6], 支持更多包括国产 SM2、SM3 在内的密码算法的同时, 也增加了新的如权限域、密钥层次结构等定义, 使得 TPM 的使用场景更加灵活广泛.

由于 TPM/TCM 广泛采用了传统密码学算法 (RSA、ECC、SM2) 作为加解密和签名原语, 量子计算机对密码算法的冲击将直接影响到 TPM/TCM 自身的安全性.

在量子计算模型下, 由于 RSA/ECC 等公钥密码算法的加解密、签名安全性被破坏, TPM/TCM 对外的远程证明能够被破解和伪造, 使得证明结果不再可信. 另外, 由于杂凑算法安全性降低, 也可能导致度量过程被破解, 无法保证下一级实体的可信, 从而导致信任链被打破. 同时存储在 TPM/TCM 芯片外部的数据也可能遭到破解.

抗量子密码算法的提出为 TPM/TCM 提供了在量子计算模型下保证安全性的前提条件. 然而如果直接在 TPM/TCM 中引入抗量子密码算法, 其不同于传统密码算法的接口和开销将会导致抗量子可信密码模块和现有可信计算软硬件体系的兼容性问题, 主要包括:

(1) 集成问题. 现有主板 BIOS、Bootloader 的信任链因算法和协议的不同无法与抗量子 TPM/TCM 芯片集成.

(2) 性能影响. 采用抗量子密码算法后, 芯片上的密码学运算、度量证明的效率也是一大问题: 受限于 TPM/TCM 安全芯片的运算速度、缓冲区大小等硬件条件, 大多数抗量子密码算法的运行速度将显著低于现有规模化商用的 RSA、ECC 等密码算法.

(3) 存储问题. 受密码算法改变影响, 同时增长的还有度量日志的长度、密钥长度, 可能会导致更多的缓存和存储问题.

(4) 其他问题. 基于 TPM2.0 规范的一些安全机制 (如依赖加密算法的增强授权等功能) 还需要进一步考虑兼容性问题.

1.2 相关工作

目前被认为具有抗量子能力的密码主要有如下类型: 基于杂凑的密码算法、基于编码的密码算法、基于格的密码算法、基于多变量的密码算法以及基于椭圆曲线同源问题的密码算法[7]. 尽管大规模量子计算机能否真正实现仍然存疑, 各国研究机构已对量子计算模型下的安全密码算法发起研究. 2017 年 NIST 开始征集后量子密码算法, 目前有 15 种算法进入决赛轮 (2020 年 7 月).

FutureTPM[8] 旨在设计和开发抗量子的可信平台模块 (TPM), 主要目标是实现从现有 TPM 环境到通过 QR 加密功能提供增强安全性的系统的平稳过渡. FutureTPM 从软件、硬件和虚拟化等不同解决方案出发, 探寻了 QR TPM 的多种实现. 主要基于 IBM TPM2.0 的开源软件实现, 并在其中添加了多种后量子算法密码学原语.

(1) 软件实现. 基于 IBM TPM2.0 的开源软件实现, 并在其中添加了对 Kyber[9]、Dilithium[10]、NTTRU[11]、L-DAA[12]、SHA3/SHAKE 等后量子算法协议的支持, 包括其密钥生成器、Hash 生成器以及 L-DAA 协议状态存储等.

(2) 虚拟化实现. 基于 QEMU 以及 libtpms 的开源实现, 并在其中添加对后量子算法的支持. 目前 QR-libtpms 已支持 Kyber、Dilithium、L-DAA, 还将向其中添加对 SPHINCS+[13]、Rainbow[14]、BIKE[15] 的支持.

(3) 硬件实现. 基于 FPGA 实现了一个具有 QR-TPM 软件实现和调度算法的最小化操作系统环境, 并通过 TCP/IP 协议栈向外提供 QR-TPM 功能.

2 抗量子可信计算体系

抗量子可信计算技术体系在架构上与传统可信体系类似, 以密码算法和协议为基础, 实现可信密码模块所需的基本功能, 可信密码模块对外提供可信根, 可信软件服务提供通用接口. 不同之处在于抗量子可信计算技术体系需要以抗量子密码算法实现相应的平台功能. 此外, 还需要考虑到传统可信体系的兼容性以及更高需求的计算性能.

我们将抗量子密码算法及安全协议集成到可信密码模块中, 为上层设备提供抗量子计算的信任根, 形成抗量子计算的身份认证、远程证明、数据保护等可信计算机制; 同时建立可信软件服务体系, 从内核层、应用层和网络层为用户应用系统提供可信软件服务. 总之, 抗量子可信计算技术体系架构主要内容包括:

(1) 面向可信计算应用的抗量子可信计算协议和安全模型.

(2) 基于抗量子密码算法的可信密码模块设计. 包括密码学系统设计、基于抗量子密码的可信计算功能扩展、模块兼容性评估等方面.

(3) 构建支持通用抗量子密码接口的可信软件服务体系.

上述体系架构沿用了传统的可信计算安全支撑平台 (TCM/TPM) 体系架构, 相关的抗量子密码算法仅作为一种新的算法被加入, 以此最小化架构上的改变对其原有安全性的影响. 该架构所具备的抗量子能力, 等同于所引入的抗量子密码算法和协议的抗量子能力.

2.1 抗量子可信密码模块扩展

抗量子可信密码模块 (QR-TCM) 是在原有硬件密码模块的基础上进行算法扩展、功能改造, 其主要结构如图 1 可信密码模块改造部分所示. 随机数发生器用于生成协议中的随机数和对称加密算法使用的密钥的运算模块; 密钥生成器包括非对称密码算法密钥生成器和 LDAA 密钥生成器, 其中非对称密码算法密钥生成器用于生成非对称密码算法密钥, LDAA 密钥生成器用于生成 LDAA 密钥; 杂凑算法引擎使用 SHA-256 或 SHAKE 杂凑算法计算消息摘要; 加解密/签名消息引擎分为非对称和对称引擎, 其中非对称密钥加解密/签名引擎为非对称密码算法提供加解密、签名、验证运算功能, 对称密钥加解密引擎为对称密码算法提供加解密功能; 非易失存储用于存储 TCM 的长期密钥 (背书密钥和存储根密钥)、完整性信息、所有者授权信息及少量重要应用数据; 易失性存储用于存储计算中产生的临时数据, 包括 PCR 寄存器、身份密钥信息以及 LDAA 签名协议状态等.

2.2 可信软件服务体系

可信软件栈是应用软件与安全芯片交互的桥梁. 安全芯片负责提供可信关键功能的硬件实现, 比如密钥生成、存储以及签名验证等, 而可信软件栈则是基于安全芯片硬件提供统一的可信计算应用程序接口, 比如密钥使用、完整性度量以及为用户应用程序建立与安全芯片的通信等.

通常在用户系统中存在两种不同的平台权限模式

定义: 内核模式和用户模式. 可信软件栈运行于用户系统中, 因此需要与用户所处的系统运行模式相匹配. 其主要分为 3 个层次, 即内核驱动层、系统服务层和用户程序层, 具体包括:

(1) 可信计算内核模块. 内核层的核心模块是安全芯片设备驱动 TDD (trusted device driver), 运行于用户平台的内核模式, 负责操作系统内核与硬件芯片之间的字节流传输, 另外还具有在电源异常等情况下保存芯片状态等功能;

(2) 可信计算核心服务. 在上述架构中 TCMlib 提供可信计算核心服务, 位于用户层, 负责与内核中芯片驱动接口函数通信时的组包、拆包、日志、审计、密钥证书的管理, 以及协调多个应用程序对安全芯片的同步访问等;

(3) 可信计算服务模块. 用户程序层的核心模块是可信服务模块 TSM, 与 TCMlib 同时运行于用户模式. 它位于可信软件栈的最高层, 直接为应用程序提供访问安全芯片的服务.

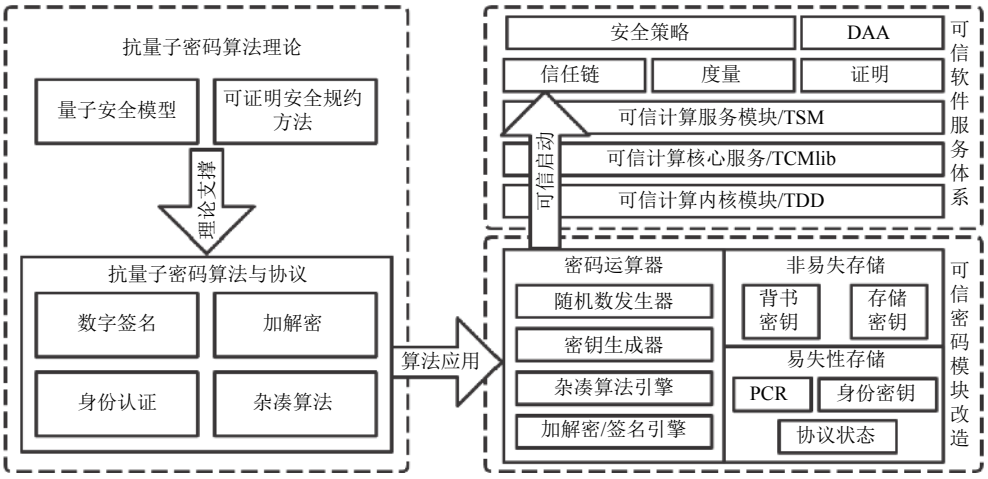

图 1 抗量子可信计算技术体系

3 抗量子可信计算安全支撑平台技术方案

3.1 平台总体架构设计

我们以抗量子可信计算体系为基础, 设计了抗量子可信计算安全支撑平台 (QR-security supporting platform for trusted computing, QR-SSPTC) 总体架构, 如图 1 右部分所示.

QR-TCM 以抗量子算法密码学原语为密码基础, 并在其上扩展与通用可信计算体系相同的安全属性和功能, 包括身份标识 (AiK)、平台数据保护 (SRK)、完整性存储与报告、资源保护以及其它辅助功能. 与 TPM 的信任链构建方法相同, 计算机通过可信启动, 将信任链从抗量子可信密码模块扩展到上层的操作系统和可信软件栈, 形成抗量子可信软件服务体系.

在平台运行过程中, 操作系统通过运行于内核层的可信计算内核模块与 QR-TCM 硬件通信, 并在用户层建立可信计算核心服务层, 提供对 QR-TCM 基础功能的封装. TCMlib 则对这些基础功能进一步封装, 以符合 QR-TCM 完整调用过程的形式提供给应用程序, 并在此基础商对密钥和会话的生命周期进行管理.

应用程序通常在被度量后启动, 以将信任链扩展到该层级. 应用程序以符合自身需求和预定义的安全策略的形式调用 TCMlib 以及下层的可信计算核心服务, 进行度量和证明. 这些需求可能是对外直接或者匿名地证明计算机平台的可信性, 或建立可信执行环境等.

3.2 QR-TCM 信任根设计

TPM2.0 标准相比于 TPM1.2 增强了灵活性, 采用主种子 (primary seed) 来在使用而非生产时生成对称密钥、非对称密钥、其它种子等关键值. 其中, 背书密钥主种子 (endorsement primary seed, EPS) 用于生成背书密钥, 是可信报告根的基础; 平台密钥主种子 (platform primary seed, PPS) 用于生成由平台固件控制的层次结构; 存储根密钥主种子 (storage primary seed, SPS) 用于生成由平台所有者控制的层次结构. 此 3 个主种子间相互独立, 但其下属的密钥层次结构允许存在交叉认证. EPS、PPS、SPS 分别对应隐私管理员、平台

固件、平台所有者 3 个不同的权限域, 每个权限域有各自的分级层次结构进行权限管理. 各管理域的主对象从相应的主种子中派生, 具有公开部分和私密部分, 其私密部分由相应主种子派生的对称密钥对其敏感区域进行加密保护. 管理域控制一个树形的层次结构, 其上的非叶子节点 (存储密钥) 都是非对称密钥, 该节点的子节点由其派生的对称密钥来进行加密保护.

现有的 TPM2.0 密码芯片具有的一个安全缺陷是没有对 EPS、PPS、SPS 进行加密保护, 并且上述三者在应用时分开管理其子密钥, 不便于密钥的统一管理. 在集中式管理的安全组织内部, 存在统一的 TPM2.0 信任根管理和控制的安全需求, 因而我们设计了安全芯片本原根密钥 (primitive root key) 的方案, 加强对 3 个保护域根密钥种子的安全管理和保护.

本原根密钥延伸了 TPM2.0 系统平台根密钥、系统背书根密钥、系统存储根密钥对其多个子密钥的树形分级结构保护方法, 在 3 个主种子上追加一个本原根密钥, 对上述 3 类根密钥进行保护, 从而加强密码芯片的安全性, 方便密钥管理.

如图 2 所示, 芯片厂商除了给芯片导入种子值外, 还需导入使用 SM2 非对称密钥算法生成的本原根密钥标识密钥, 该密钥存储于片内, 并在芯片全生命周期中有效. 本原根密钥来自于芯片外部, 由芯片厂商在制造时导入芯片的内部非易失存储, 为国密 SM2 算法生成的非对称密钥, 其密钥对不会以任何形式存储于片外, 并受到芯片物理保护.

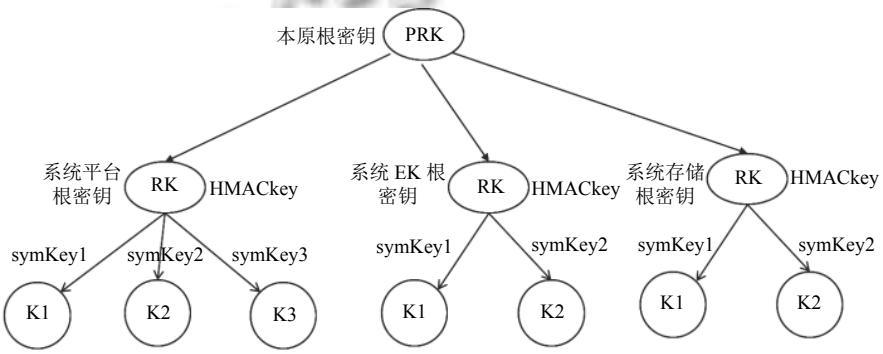

图 2 本原根密钥

芯片针对平台固件、平台所有者、隐私管理员分为 3 个不同的权限域, 对应平台主种子 (PPS)、存储主种子 (SPS)、背书主种子 (EPS) 3 类种子值派生的密钥层次结构, 其中 PPS 生成平台根密钥、SPS 生成存储根密钥、EPS 生成背书密钥. PPS、SPS、EPS 在片内存储时, 受本原根密钥标识密钥通过 SM3 KDF 派生密钥算法导出的 SM4 完整性保护密钥、机密性保护密钥的加密保护.

在使用相应主种子值生成派生根密钥 (非对称) 时, 需要对应权限域的授权值或者授权策略, 包括 PPS 的平台授权, SPS 的所有者授权和 EPS 的背书授权三者之一, 以及本原根密钥标识密钥的授权. 在平台对主种子进行更换操作时, 包括平台对 PPS、EPS、SPS 的更改, 平台所有者更改导致的 SPS 更改等, 平台需要取得本原根密钥标识密钥的授权, 并在更换种子值后由本原根密钥标识密钥导出的保护密钥重新对种子值进行加密保护.

3.3 TCMlib 功能模块

与 TCM 直接交互的是位于操作系统内核层的 TCM 设备驱动. 操作系统内核通过总线与 TCM 间的函数调用需要满足标准中繁琐的步骤和要求. 而 TCMlib 位于操作系统内核上层, 将这些函数调用进一步封装, 为上层应用提供调用 QR-TCM 的接口; 并通过对上层暴露更少的接口, 减少 TCM 可能存在的攻击面, 使得更上层的 TCM 服务模块 TSM 对 TCM 芯片的调用更加简单和安全. 此外, TCMlib 还具有 TCM 管理、密钥管理、完整性度量与证明的关键功能.

(1) TCM 管理

TCM 管理指对 TCM 本身状态、数据结构、基础功能支撑等的管理. 具体包括 TCM 错误状态、TCM 的启动状态、PCR 寄存器状态; TCM 密钥、会话、缓存等的定义; TCM 缓存构建、哈希操作、传输命令、获取错误信息等. TCMlib 将 TCM 管理功能进行封装, 为上层提供包括 TCM 启动、物理使能、激活、获取

所有权、读取背书密钥、进行会话、创建密钥、加载密钥、获取密钥、签名、PCR 读取和拓展等功能的操作.

(2) 密钥管理

由于 TCM 芯片内部 NV 是有限的, 用户和上层应用创建的密钥只有部分会存储于芯片内部, 而其它密钥会以密文的形式存储于芯片外部. 这部分加密后的密钥需要由 TCMlib 进行管理, 对每个密钥进行标识. 用户从保存于芯片内部的可信存储根中导出对称密钥, 保护下一级存储于芯片外部的子密钥. 子密钥可以逐级生成, 形成树形保护结构.

(3) 完整性度量和证明

TCMlib 提供完整的对称加解密算法以及非对称加解密、签名算法. 一方面, TCMlib 基于这些算法完成对应用程序的度量和证明; 另一方面, 操作系统在启动时会将信任链传递到 TCMlib, 而 TCMlib 则负责度量上层应用, 将信任链传递到更上层.

上述 TCM 的核心安全功能是由 TCMlib 通过调用下层的 TCM 芯片内核驱动接口实现的. 而 TCMlib 向上提供的主要接口命令包括 TCM 管理命令、会话命令、密钥相关命令、PCR 操作命令、支撑命令. TCM 管理命令为对 TCM 自身状态的设置, 包括启动、所有者权限获取和清除等操作; 会话命令控制授权协议发起的类型, 并确定何时终止会话释放资源; 密钥相关命令包括获取平台公钥、导入导出密钥和使用指定密钥签名等; PCR 操作命令包括读取 PCR 寄存器中的值以及向 PCR 中增加度量值的功能; 支撑命令包括对具体数据结构的操作以及对文件或字符串的杂凑、控制日志或调试信息的输出等功能.

3.4 可信密码模块功能改造

(1) XMSS 远程证明

扩展的 Merkle 签名方案 (XMSS) 是一种基于哈希的数字签名系统, IETF 在 RFC 8391 中将其标准化[16]. XMSS 提供加密的数字签名, 但不直接依赖数学困难问题. 它的安全性仅依赖于哈希函数的属性. 我们将 XMSS 应用到 QR-TCM 以实现远程证明的抗量子扩展.

远程证明过程中的角色分为可信计算平台和远程验证方, 其中可信计算平台包含 TPM/TCM 安全芯片和主机平台, 它们共同完成平台完整性证明. 主机平台主要是对平台的完整性进行度量和报告, 而安全芯片 TPM/TCM 主要是完成远程证明的签名过程.

远程验证方是远程证明的挑战方, 它请求可信计算平台证明当前系统的完整性状态, 并根据相应的验证策略验证可信计算平台的完整性日志和安全芯片签名. 远程证明之前, 远程验证方还必须知道可信计算平台的 AIK 公钥和可信第三方的公钥, 此处远程证明的公私钥即为 XMSS 的公私钥.

在远程证明的核心步骤是远程验证方对可信计算平台发起挑战, 平台证明当前运行状态的完整性作为证明应答. 具体步骤如图 3 所示.

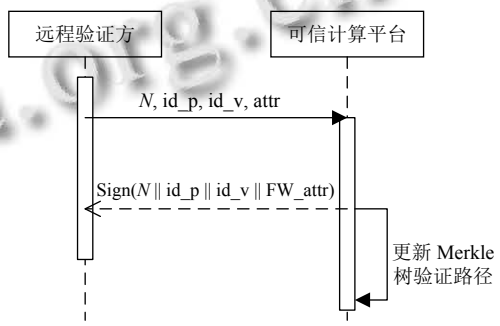

图 3 XMSS 远程证明过程

1) 远程验证方预先持有可信计算平台的签名公钥. 远程验证方向可信计算平台发送随机数 N, 自身身份标识 id_v, 对方身份标识 id_p, 以及需要验证的属性标识 attr.

2) 可信计算平台验证随机数 N 的新鲜性, 并验证远程验证方发送的 id_v、id_p 及 attr 是否合法. 验证通过后使用 Merkle 树中某个未使用过的叶子节点的签名密钥, 即 XMSS 签名私钥对 N、id_v、id_p 以及相应属性证明结果 FW_attr 进行签名并发送. 此后, 可信计算平台需要预先选择新的叶子节点签名密钥, 并预备其证明路径.

3) 远程验证方收到可信计算平台发回的相应后, 使用 XMSS 签名公钥对签名结果进行验签, 并验证 id_p、id_v 和 attr 的正确性. 验证通过则 FW_attr 为可信计算平台对远程验证方真实的远程证明结果.

(2) LDAA

直接匿名证明 (DAA)[17] 是一种远程证明协议, 使 TPM 和主机 IoT 设备不仅可以向其它设备提供身份验证信息, 证明其处于可信状态, 并且还能保证验证方不能获取受验证设备的其它信息. LDAA 是基于格的直接匿名证明方案, El Bansarkhani 等在原有的 DAA 基础上设计了 LDAA[18], 而 El Kassem 等人[19] 进一步为 TPM 等

资源受限环境优化了 LDAA 协议.

前述远程证明过程无法保证可信计算平台不向远程验证方暴露除验证信息之外的其它隐私信息, 比如 id_p. 为此, 直接匿名证明协议基于零知识证明提供了一个在不暴露可信计算平台隐私的前提下向远程验证方提供证明的方案. 然而已在 TPM/TCM 中应用的 DAA 协议都是基于 RSA 或 ECC 的, 对于量子计算模型下的攻击无法抵御.

LDAA 是基于格问题提出的 DAA 方案, 能够在量子计算模型下为 DAA 协议提供一定的防护. 其基本角色分为 TCM、主机、凭证颁发方、远程验证方, DAA 过程主要分为加入 (join)、签名 (sign)、验证 (verify) 3 个步骤. 我们的抗量子可信计算安全支撑平台将集成通用 LDAA 协议方案, 为可信计算平台对外提供高安全性的平台匿名身份认证功能, 确保即使攻击者截获直接匿名证明消息, 通过离线的攻击性分析和破解, 也无法破坏可信计算平台的匿名性和不可链接性.

4 系统实现与分析

4.1 方案架构

QR-TCM 硬件测试平台采用树莓派 4B 作为 QR-TCM 模拟测试设备, 其方案架构如图 4 所示.

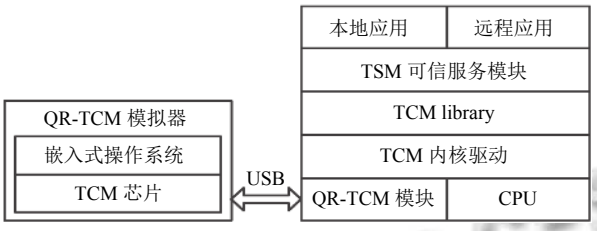

图 4 QR-TCM 硬件测试平台方案架构

QR-TCM 通过 USB 接口与主机平台进行通信, 两者建立 RNDIS (remote network device interface specification, 远程网络设备接口规范) 链接, 主机平台或者 USB 设备自身通过调用 QR-TCM 模块实现各项安全功能的应用. 在 RNDIS 连接上使用 TCP/IP 协议, USB 设备运行一个轻量级 DHCP 服务器, 它动态地为主机分配一个 IP 地址. 通过这个 IP 链路, 基于 USB 的可信设备可以与网络上的其他机器进行通信. 系统软件体系结构如下:

(1) USB 驱动程序. 驱动程序允许 USB 可信计算平台在用户终端前显示为 IP 网络设备. 我们在内核模块中使用了比较通用的 Linux USB Ethernet/RNDIS gadget 驱动程序.

(2) TCM 模拟器抗量子扩展. 基于 TCM 硬件芯片在模拟器平台上扩展本原根密钥方案、XMSS 算法以及 L-DAA 协议.

(3) QR-TCM 内核驱动. 位于主机平台的操作系统内核中, TCM 驱动程序维护设备文件的 I/O 状态, 提供 tcmSend 和 tcmReceive 操作来发送和接收来自 QR-TCM 模块的数据.

(4) TCMlib 库和 TSM 服务. 位于主机平台的用户层, 向应用程序提供封装的函数功能接口和操作系统服务, 并与 QR-TCM 内核驱动通信.

(5) 本地应用和远程应用, 调用 QR-TCM 所提供的可信计算服务, 例如随机数生成、密钥管理、远程证明等.

4.2 QR-TCM 设备信任根

本方案的 QR-TCM 设备可信基 (TCB) 由 3 部分组成: QR-TCM 芯片, 抗量子密码库 QR-TCMlib 以及内核设备驱动. QR-TCM 芯片实现关键的抗量子安全功能; 抗量子密码库 QR-TCMlib 以芯片为基础, 为应用程序提供安全功能的接口和部分辅助功能 (如: 与芯片交互的数据包的构建、抗量子的 hash 函数的实现).

我们为抗量子密码库 QR-TCMlib 定义了通用的接口规范, 相关安全功能全部采用标准的可信计算接口, 如: 启动 QR-TCM 芯片 QRTCM_Startup, 会话创建 APopen, 关闭会话并中止授权协议 APclose. 同时, 制定了所有密码功能的接口规定, 如: 把一个受保护的密钥导入 TCM 并分配密钥句柄 uint32_t QRTCM_LoadKey_internal (apsess* sess, uint32_t keyhandle, QRTCM_KEY *keyparms, uint32_t *newhandle), 获取一个已经载入到 TCM 中的非对称密钥的公钥 uint32_t QRTCM_GetPubKey_internal (apsess *sess, uint32_t keyhandle, QRTCM_PUBKEY *tcmdata). 所有应用与芯片交互的数据包均采用 tcm_buffer 进行封装, 以规范数据格式.

4.3 QR-TCM 设备信任根

抗量子可信计算体系在上述 QR-TCM 设备信任根的基础上, 将信任逐步传递给安全功能接口, 为用户应用提供所需的抗量子安全服务, 并为应用程序与安全芯片之间的数据传输提供保护. 安全功能所需的密码模块主要由芯片和密码库实现, 向用户应用开放功

能接口, 如: 提供安全执行环境 (可信引导、安全存储、可信执行), 远程证明 (完整性度量和证明, 软件证明).

我们在远程证明功能中提供了抗量子数字签名算法 XMSS. 步骤如下: (1) 开启 TCM. (2) 创建主会话 APopen. (3) QRTCM_CreateXMSSKey 创建密钥. (4) 与芯片交互, QRTCM_LoadXMSSKey_internal 将密钥导入 QR-TCM 并分配密钥句柄. (5) 创建签名会话 QRTSS_XMSS_APopen. (6) QRTCM_XMSS_Sign 进行 XMSS 抗量子签名, 该过程首先将构造好的命令包 (包括密钥句柄、需签名的数据) 发送给芯片. 由芯片进行验证, 验证成功后取出数据以及密钥信息, 进行签名操作. 然后, 构建签名应答包返回给用户程序. (7) 用户程序验证应答包可信性, 取得 XMSS 签名. (8) 关闭签名会话 QRTSS_XMSS_APclose. (9) 关闭主会话 APclose. 至此, XMSS 签名过程完成. 本方案在 QR-TCMlib 密码库中实现了 XMSS 的验证功能, 以便没有芯片的终端进行签名验证操作.

所有命令包与应答包都严格按照 TCM 命令字节码接口进行规范封装, 如表 1 和表 2.

表 1 输入数据格式

长度 (B)	类型	名称	描述
2	QRTCM_TAG	标识	0x00C2
4	UNIT32	数据长度	输入数据总字节数
4	QRTCM_COMMAND_CODE	命令码	QRTCM_ORD_Xmss_command
4	QRTCM_KEY_HANDLE	签名密钥句柄	无
4	UNIT32	待签名数据长度	无
32	BYTE[]	待签名数据	无

表 2 输出数据格式

长度 (B)	类型	名称	描述
2	QRTCM_TAG	标识	0x00C5
4	UNIT32	数据长度	输出数据总字节数
4	QRTCM_RESULT	返回码	QRTCM_ORD_Xmss_response
4	UNIT32	签名数据长度	无
<>	BYTE[]	签名数据	无

4.4 原型系统实现与评估

在抗量子可信计算的系统中, 我们在硬件层采用了 TCM 芯片, 并实现了抗量子密码库 QR-TCMlib, 通过内核设备驱动将二者联系起来, 共同构成 QR-TCM 设备信任根. 在此基础上, 系统为用户提供各种抗量子可信计算功能, 如: 以 XMSS 为基础的远程服务、封装和存储加密、完整性度量和证明以及 L-DAA.

我们对抗量子可信计算原型系统进行了性能评估, 主要是签名认证、远程证明的性能测试. 测试系统硬件配置如下: 主机 CPU 为 2.8 GHz Intel(R) Core(TM) i7-7700HQ、8 GB RAM, 操作系统 Ubuntu 16.04.1. QR-TCM 模拟器平台 Raspberry Pi 4b, CPU 为 1.5 GHz Broadcom BCM2711、4 GB RAM. 对 XMSS 远程接口 (如密钥生成时间、签名时间和验证时间) 进行了 50 次反复测试, 在实验环境中, 生成高度为 10 的 Merkle 树签名密钥花费 1.80 s; 签名花费 1.85 s, 除去生成密钥的时间实际签名花费 0.05 s; 签名验证仅需 0.000 89 s; 签名长度为 2 532 B. 在计算能力、存储能力快速增长的应用环境下, 这样的开销是可以接受的.

4.5 横向评估

(1) FutureTPM

我们在第 1.2 节中已对 FutureTPM 进行过介绍. FutureTPM 的公开文档中给出了抗量子算法的部分性能测试结果. 在其虚拟化实现上 FutureTPM 移植并使用 Dilithium 算法进行的签名运算, 单次签名时长为 0.265 s[20], 是 QR-TCM 模拟器的 5 倍.

对于该结果, 一种合理的解释是, FutureTPM 的最终设计目标是硬件 TPM 模块, 并且其虚拟化实现基于 QEMU[21], 因此有限的缓存设计以及 LLVM[22] 的解释执行使计算性能受到限制. 目前为止 FutureTPM 没有对外公开更详细的性能数据.

(2) TPM-Based PQC

TPM-Based PQC[23] 将后量子算法集成至 Mbed TLS 密码算法库中, 基于现有的 TPM 以及后量子密码算法, 通过对 TPM 可信硬件的利用和对 TLS 协议的抗量子改造, 为 IoT 设备提供抗量子交叉认证能力.

TPM-Based PQC 采用的签名算法是 SPHINCS+. 该算法以候补候选算法进入 NIST 第 3 轮后量子算法评估.

TPM-Based PQC 在 Raspberry Pi 3b 上对密码学原语进行了性能测试. 其 SW(SHA2-256)+TPM(TRNG) 的签名/验签场景实验结果对本实验最具有参考价值. 在该场景下, 实验平台单次签名花费 0.230 s, 签名验证花费 0.013 s. 与该结果相比, QR-TCM 模拟器上签名耗

时长一倍, 但签名验证时间更小. 导致该结果的原因除算法差异外, 还因为 QR-TCM 实验结果包含了主机平台对 QR-TCM 的调用时延, 而 TPM-Based PQC 仅在其实验平台本身上进行密码计算.

(3) Streaming SPHINCS+

Streaming SPHINCS+[24] 为资源受限设备优化了后量子签名算法 SPHINCS+的实现, 采用流式传输接口降低了内存需求. 该方案基于微软开源的软件 TPM 实现, 并在 ARM Cortex-M4 平台上进行了性能测试. 对于性能开销最低的场景, 实验平台单次签名花费 7.54 s, 签名验证花费 2.61 s.

该结果显著高于 QR-TCM 的时间开销, 原因是该方案选择了低功耗嵌入式实验平台来模拟实际密码芯片, 其主频仅有 168 MHz.

(4) Software QR-TPM

Software Emulation QR-TPM[25] 参考了 FutureTPM 的软件实现, 并在其上评估了移植后量子算法 Kyber、NTTRU、Dilithium 以及 L-DAA 协议对于 TPM 的影响. 方案在 i5-5257U 平台上实验了 QR-TPM 软件模拟器并进行了密码学原语性能测试. 实验平台上使用 Dilithium 算法单次签名花费 0.185 s, 签名验证花费 0.175 s.

该方案选择了 Intel 移动平台 CPU, 其计算性能优于其它实验平台, 因此具有最短的时间开销, 但对密码芯片和移动嵌入式领域的参考价值较低. 此外该时间开销仅表明密码学原语性能, 不包括模拟硬件 TPM 所导致的时间开销. 表 3 总结了关于签名性能的横向评估结果.

表 3 签名性能横向评估结果

方案	后量子签名算法	签名时间开销 (s)	验证时间开销 (s)
本文	XMSS	0.50	0.000 89
FutureTPM 虚拟化实现	Dilithium	0.265	—
TPM-Based PQC	SPHINCS+	0.230	0.013
Streaming SPHINCS+	SPHINCS+	7.54	2.61
Software QR-TPM	Dilithium	0.185	0.175

表 3 的时间开销对比结果证明我们的原型系统能够在实际使用中提供良好的可用性. 需要说明的是平台的抗量子能力即由这些后量子算法提供. 传统算法如 RSA 数字签名算法无法抵御大型量子计算机的攻击, 原因在于其基于的数学困难问题“大数分解难题”能够被 Shor 算法在多项式时间内破译[1], 即能够通过签名算法公钥计算出其私钥, 使得签名结果不再可信. 而后量子算法如 XMSS 的困难性仅依赖于其所使用杂凑函数的安全性, 在量子计算机上不能被快速求解, 因此其签名结果具有抗量子能力.

5 结语

量子计算机的快速推进, 对可信计算技术, 特别是 TCM/TPM 的公钥密码算法和协议产生了严重安全影响, 基于大数分解问题或离散对数问题等困难问题的公钥密码算法与协议如 RSA、ECC、DH 密钥交换等在量子计算模型下能够被快速求解, 其安全性将被大幅降低. 目前世界各研究机构正在对抗量子密码算法进行研究和评估.

我们根据可信计算技术体系, 结合抗量子密码算法和协议, 提出了一个通用的抗量子可信计算体系; 设计并初步实现了一个基于 TCM 硬件芯片的抗量子扩展 QR-TCM, 具有 TCM 的通用功能, 并通过在其上移植抗量子密码算法和协议, 使其具有抵抗量子算法攻击的安全性, 并且具有可以接受的性能开销. 本研究通过对可信密码模块进行软件扩展, 使其在不显著降低安全性的情况下增强抵抗量子攻击能力, 并为未来硬件实现的 QR-TCM 芯片提供了技术实现参考.

参考文献

1 Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 1999, 41(2): 303–332. [doi: 10.1137/S0036144598347011]

2 Arute F, Arya K, Babbush R, *et al*. Quantum supremacy using a programmable superconducting processor. Nature, 2019, 574(7779): 505–510. [doi: 10.1038/s41586-019-1666-5]

3 Grover LK. Quantum mechanics helps in searching for a needle in a haystack. Physical Review Letters, 1997, 79(2): 325–328. [doi: 10.1103/PhysRevLett.79.325]

4 Bennett CH, Bernstein E, Brassard G, *et al*. Strengths and weaknesses of quantum computing. SIAM Journal on Computing, 1997, 26(5): 1510–1523. [doi: 10.1137/S0097539796300933]

5 国家密码管理局. GB/T 29829-2013 信息安全技术 可信计算密码支撑平台功能与接口规范. 北京: 中国标准出版社, 2014.

6 Trusted Computing Group. Trusted Platform Module Library Specification, Family 2.0, Level 00, Revision 01.16. October 2014. https://trustedcomputinggroup.org/resource/tpm-library-specification/. (2019-11-08)[2021-11-04].

7 郁昱. 后量子密码专栏序言. 密码学报, 2017, 4(5): 472–473.

8 Future TPM. Future proofing the connected world: A quantum-resistant trusted platform module. European Union's Horizon 2020 research and innovation programme under grant agreement No.779391. 2017. https://futuretpm.eu/index.php/results/public-rtd-deliverables. (2021-02-09)[2021-11-04].

9 Bos J, Ducas L, Kiltz E, *et al*. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. 2018 IEEE European Symposium on Security and Privacy (EuroS&P). London: IEEE, 2018. 353–367.

10 Ducas L, Kiltz E, Lepoint T, *et al*. Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018, (1): 238–268.

11 Lyubashevsky V, Seiler G. NTTRU: Truly fast NTRU using NTT. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019, (3): 180–201.

12 El Kassem N. Lattice-based direct anonymous attestation [Ph.D. Dissertation]. Guildford: University of Surrey, 2020.

13 Bernstein DJ, Hülsing A, Kölbl S, *et al*. The SPHINCS$^+$ signature framework. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London: ACM, 2019. 2129–2146.

14 Ding JT, Schmidt D. Rainbow, a new multivariable polynomial signature scheme. Third International Conference on Applied Cryptography and Network Security. New York: Springer, 2005. 164–175.

15 Aragon N, Barreto P, Bettaieb S, *et al*. BIKE: Bit flipping key encapsulation. https://hal.archives-ouvertes.fr/hal-01671903. (2017-12-22)[2021-11-04].

16 Hüelsing A, Butin D, Gazdag SL, *et al*. XMSS: eXtended Merkle signature scheme. RFC 8391, Internet Research Task Force, 2018.

17 Brickell E, Camenisch J, Chen LQ. Direct anonymous attestation. Proceedings of the 11th ACM Conference on Computer and Communications Security. Washington: ACM, 2004. 132–145.

18 El Bansarkhani R, El Kaafarani A. Direct anonymous attestation from lattices. IACR Cryptology ePrint Archive: Report 2017/1022, 2017.

19 El Kassem N, Chen L, El Bansarkhani R, *et al*. More efficient, provably-secure direct anonymous attestation from lattices. Future Generation Computer Systems, 2019, 99: 425–458.

20 FutureTPM. D5.4 Report on implementation. FutureTPM, 2020. https://futuretpm.eu/images/Deliverables/FutureTPM-D54-Report-on-implementation-PU-M30.pdf. (2020-07-14)[2021-11-04].

21 Bellard F. QEMU, a fast and portable dynamic translator. Proceedings of the Annual Conference on USENIX Annual Technical Conference. Anaheim: ACM, 2005. 41.

22 Lattner C, Adve V. LLVM: A compilation framework for lifelong program analysis & transformation. International Symposium on Code Generation and Optimization, 2004. CGO 2004. San Jose: IEEE, 2004. 75–86.

23 Paul S, Schick F, Seedorf J. TPM-based post-quantum cryptography: A case study on quantum-resistant and mutually authenticated TLS for IoT environments. The 16th International Conference on Availability, Reliability and Security. Vienna: ACM, 2021. 3.

24 Niederhagen R, Roth J, Wälde J. Streaming SPHINCS+ for embedded devices using the example of TPMs. IACR Cryptology ePrint Archive: Report 2021/1072, 2021.

25 Fiolhais L, Martins P, Sousa L. Software emulation of quantum resistant trusted platform modules. ICETE, 2020, (2): 477–484.