ICS 35.030
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0011—2023
代替 GM/T 0011—2012

可信计算
可信密码支撑平台功能与接口规范

Trusted computing—Trusted computing functionality and interface specification of cryptographic support platform

2023-12-04 发布　　2024-06-01 实施

国家密码管理局　发布

目　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则　第 1 部分:标准化文件的结构和起草规则》的规定起草。

本文件代替 GM/T 0011—2012《可信计算 可信密码支撑平台功能与接口规范》,与 GM/T 0011—2012 相比,除结构调整和编辑性改动外,主要技术内容变化如下:

a) 删除了“术语和定义”中的部件(见 2012 年版的 3.1.1)、对象(见 2012 年版的 3.1.3)、密码模块密钥(见 2012 年版的 3.1.12)、密钥管理中心(见 2012 年版的 3.1.13)、平台身份密钥(见 2012 年版的 3.1.15)、平台加密密钥(见 2012 年版的 3.1.16)、双证书(见 2012 年版的 3.1.17)、实体(见 2012 年版的 3.1.18);

b) 增加了“术语和定义”中的背书密钥(见 3.9)、授权值 (见 3.14)、授权策略 (见 3.15)、上下文(见 3.16)、背书授权(见 3.17)、平台固件(见 3.18)、主密钥(见 3.19)、主对象(见 3.20)、主种子(见 3.21)、存储密钥(见 3.23);

c) 删除了缩略语 NV、PIK 和 PEK(见 2012 年版的 3.2);

d) 增加了缩略语 AK、CRTM、DRTM、ECDH、EPS、GPIO、HMAC、IPL、KDF、MBR、PP、PPS、RNG、RTM、RTR、RTS、SPS、SRTM、TBB、TCB 和 UEFI(见第 4 章);

e) 更改了“缩略语”EK 的名称(见第 4 章);

f) 更改了“平台体系架构”和“功能原理”的一些内容(见 6.1 和 6.2,2012 年版的 4.1 和 4.3);

g) 删除了“密码算法要求”(见 2012 年版的 4.2);

h) 删除了“可信计算密码支撑平台功能接口”(见 2012 年版的第 5 章);

i) 增加了“可信计算密码支撑平台概述”(见第 5 章);

j) 增加了“可信计算密码支撑平台功能”(见第 6 章);

k) 增加了“可信密码模块接口”(见第 7 章);

l) 增加了 SM2 非对称加解密的指令实现要求(见 7.7);

m) 增加了“可信密码模块证实方法”(见第 8 章);

n) 删除了规范性附录 A、附录 B 和附录 C(见 2012 版附录 A、附录 B 和附录 C);

o) 增加了规范性附录 A(见附录 A),定义了命令码、返回码、常量和数据结构。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位:国民技术股份有限公司、中国科学院软件研究所、北京信息科技大学、联想(北京)有限公司、中国电子技术标准化研究院、武汉大学、北京大学、北京奇虎科技有限公司、大唐高鸿信安(浙江)信息科技有限公司、中电科技(北京)有限公司、神州网信技术有限公司、浪潮电子信息产业股份有限公司、兴唐通信科技有限公司、阿里云计算有限公司、深圳数字电视国家工程实验室股份有限公司、国家计算机网络与信息安全管理中心、公安部第三研究所、国民认证科技(北京)有限公司、北京蚂蚁云金融信息服务有限公司、华为技术有限公司、郑州迪维勒普科技有限公司、北京卓识网安技术股份有限公司、同方股份有限公司、长春吉大正元信息技术股份有限公司、联想(北京)信息技术有限公司、新华三技术有限公司、中电科网络安全科技股份有限公司、无锡江南信息安全工程技术中心、中国人民解放军国防科学技术大学。

本文件主要起草人:秦宇、刘鑫、付月朋、刘大遒、吴秋新、韦卫、李汝鑫、张严、王惠莅、孙彦、王娟、严飞、沈晴霓、张晓磊、张屹、郑驰、张佳建、陈小春、孙亮、王强、杨尚欣、吴保锡、白欣璐、王悦、付颖芳、

肖鹏、李新国、岳志军、王晖、陶源、柴海新、李俊、初晓博、张小虎、张梦良、许东阳、刘韧、刘锋、姚金龙、吴会军、杜克宏、卢卫疆、冯伟、李为、张立强、余发江、赵波、李业旺、秦文杰、罗武。

本文件及所代替文件的历次版本发布情况为：

——2012 年首次发布 GM/T 0011—2012；

——本次为第一次修订。

引　言

为满足可信计算产业不断发展的新需求，本文件以密码算法应用为核心，以可信计算技术应用需求为基础，描述了可信计算密码支撑平台的功能，参考了我国密码算法在国际上可信计算标准中的采纳情况及我国可信计算技术、国际上可信计算技术的应用成果，定义了可信计算密码支撑平台接口形式。本文件符合不同应用场景下可信计算密码支撑平台设计需求，兼容各种硬件平台、宿主机软件系统、应用系统，确保产业界产品的统一性和兼容性，用于指导我国可信计算相关产品开发和应用。

可信计算
可信密码支撑平台功能与接口规范

1 范围

本文件给出可信计算密码支撑平台的体系框架和功能原理，规定了可信密码模块的接口规范，描述了对应的证实方法。

本文件适用于可信计算密码支撑平台相关产品的研制、生产、测评与应用开发。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 20518 信息安全技术 公钥基础设施 数字证书格式
GB/T 25069 信息安全技术 术语
GB/T 32905 信息安全技术 SM3 密码杂凑算法
GB/T 32907 信息安全技术 SM4 分组密码算法
GB/T 32915 信息安全技术 二元序列随机性检测方法
GB/T 32918.2 信息安全技术 SM2 椭圆曲线公钥密码算法 第 2 部分：数字签名算法
GB/T 32918.3 信息安全技术 SM2 椭圆曲线公钥密码算法 第 3 部分：密钥交换协议
GB/T 32918.4 信息安全技术 SM2 椭圆曲线公钥密码算法 第 4 部分：公钥加密算法
GB/T 35276 信息安全技术 SM2 密码算法使用规范
GM/T 0012 可信计算 可信密码模块接口规范
GM/T 0058 可信计算 TCM 服务模块接口规范
GM/Z 4001 密码术语

3 术语和定义

GB/T 25069 和 GM/Z 4001 界定的以及下列术语和定义适用于本文件。

3.1

存储主密钥 storage master key

用于保护操作系统密钥和用户密钥的主密钥。

3.2

可信计算平台 trusted computing platform

构建在计算系统中，用于实现可信计算功能的支撑系统。

3.3

可信计算密码支撑平台 cryptographic support platform for trusted computing

可信计算平台的重要组成部分，包括密码算法、密钥管理、证书管理、密码协议、密码服务内容，为可信计算平台自身的完整性、身份真实性和数据保密性提供密码支持。

3.4

可信度量根　root of trust for measurement

一个可信的完整性度量单元，是可信计算平台内进行可信度量的基础。

注：可信度量根由硬件、固件、软件等方式实现或组合构成。

3.5

可信存储根　root of trust for storage

存储主密钥，是可信计算平台内进行可信存储的基础。

3.6

可信报告根　root of trust for reporting

可信计算密码模块密钥，是可信计算平台内进行可信报告的基础。

3.7

可信密码模块　trusted cryptography module

可信计算平台中，提供密码运算功能，具有受保护存储空间的密码模块。

3.8

TCM 服务模块　TCM service module

可信计算密码支撑平台外部的软件模块，提供访问可信密码模块的软件服务接口。

3.9

背书密钥　endorsement key

可信密码模块中的一个非对称密钥，不可以迁移，生成于平台的生产过程，代表着每个平台的身份。

3.10

平台配置寄存器　platform configuration register

可信密码模块内部用于存储平台完整性度量值的存储单元。

3.11

完整性度量　integrity measurement

采用密码杂凑算法对被度量对象计算其杂凑值的过程。

3.12

完整性度量值　integrity measurement value

组件被度量后得到的杂凑值。

3.13

信任链　trusted chain

在计算系统启动和运行过程中，使用完整性度量方法在组件之间所建立的信任传递关系。

3.14

授权值　authentication value

用于访问授权的八比特的字节字符串，用作口令或者派生出一个用于 HMAC 计算的密钥。

[来源：ISO/IEC 11889-1：2015，3.2]

3.15

授权策略　authentication policy

执行策略命令所生成的、并用于访问授权的杂凑值。

[来源：ISO/IEC 11889-1：2015，3.3]

3.16

上下文　context

提供有关数据对象访问信息的数据集合，用来将数据对象与同一类型的其他数据区分开、或将数

据对象的不同版本区分开。

[来源:ISO/IEC 11889-1:2015,3.9]

3.17

背书授权 endorsement authorization

使用背书认证或者背书策略完成的授权。

[来源:ISO/IEC 11889-1:2015,3.20]

3.18

平台固件 platform firmware

系统引导和平台初始化操作所需的代码,由平台制造商添加到平台中。

注:通常但不完全特指BIOS或UEFI代码。

[来源:ISO/IEC 11889-1:2015,3.48]

3.19

主密钥 primary key

从与内部层次结构关联的主种子创建的主密钥。

[来源:ISO/IEC 11889-1:2015,3.49]

3.20

主对象 primary object

主密钥或具有敏感区域的数据区,该数据区使用从对象的公共区域和主种子创建的对称密钥加密。

[来源:ISO/IEC 11889-1:2015,3.50]

3.21

主种子 primary seed

包含在从中派生主密钥和主对象的 TCM 中的大随机值。

[来源:ISO/IEC 11889-1:2015,3.52]

3.22

可信根 root of trust

始终以预期方式运行的单元。是系统的可信基点,也是实施安全控制的基点。

注:完整的可信根集至少具有一组最小的功能,以便能够描述影响平台可信度量的平台特性。

[来源:ISO/IEC 11889-1:2015,3.59,有修改]

3.23

存储密钥 storage key

用于保护存储在 TCM 外部对象的子密钥。

[来源:ISO/IEC 11889-1:2015,3.72,有修改]

4 缩略语

下列缩略语适用于本文件。

AK:证明密钥(Attestation Key)

CRTM:可信度量根核(Core Root of Trust for Measurement)

DRTM:动态可信度量根(Dynamic Root of Trust Measurement)

ECDH:基于椭圆曲线的密钥交换(Elliptic Curve Diffie-Hellman)

EK:背书密钥(Endorsement Key)

EPS:背书密钥主种子(Endorsement Primary Seed)

GPIO:通用输入输出(General Purpose I/O)
HMAC:采用带密钥杂凑的消息鉴别码(Keyed-hash Message Authentication Code)
IPL:初始化程序加载器(Initial Program Loader)
KDF:密钥导出函数(Key Derivation Function)
MBR:主引导分区(Master Boot Record)
NV:非易失性 (Non—Volatility)
PCR:平台配置寄存器(Platform Configuration Register)
PP:物理存在(Physical Presence)
PPS:平台主种子(Platform Primary Seed)
RNG:随机数发生器(Random Numeral Generator)
RTM:可信度量根(Root of Trust for Measurement)
RTR:可信报告根(Root of Trust for Reporting)
RTS:可信存储根(Root of Trust for Storage)
SPS:存储主种子(Storage Primary Seed)
SMK:存储主密钥(Storage Master Key)
SRTM:静态可信度量根(Static Root of Trust Measurement)
TBB:可信构件(Trusted Building Block)
TCB:可信计算基(Trusted Computing Base)
TCM:可信密码模块(Trusted Cryptography Module)
TSM:TCM 服务模块(TCM Service Module)
UEFI:统一可扩展固件接口(Unified Extensible Firmware Interface)

5 可信计算密码支撑平台概述

5.1 可信计算

可信计算是一种旨在增强计算机系统可信性的综合性信息安全技术。可信技术在计算机系统中,建立一个可信根,从可信根开始到硬件平台、到操作系统、再到应用,逐级度量,逐级认证,逐级信任,把这种信任扩展到整个计算机系统,并采取防护措施,确保计算资源的完整性和计算行为的可预期性,从而提高计算机系统的可信性。

5.2 可信构件

一个可信构件是由一个或一组可完成可信根实例化的部件组成。

5.3 可信计算基

可信计算基(TCB)是负责维持系统安全策略的系统软硬件资源。TCB 的一个重要属性是它应防止自身被任何不属于 TCB 的硬件或软件所破坏。

5.4 可信边界

可信构件(TBB)和可信根组成了可信边界,在其中包括了对计算机的最小配置进行完整性度量、存储和报告的实体。在更复杂的系统里,应基于可信度量根核(CRTM),对其他代码进行可信度量,并且在平台配置寄存器(PCR)记录度量结果,将信任扩展到其他代码。

5.5 可信传递

可信传递是基于可信根逐个对被度量的模块进行可信度量并扩展信任链的过程。

可信传递应通过 TCM 支持以下两种方法之一来实现：

a) 按照安全策略执行的一个操作，以允许后续的操作应获得 TCB 的控制权；

b) 对于后续操作的度量，建立独立评估的信任关系。

5.6 可信授权

当可信度量根(RTM)开始执行可信度量根核(CRTM)时，由可信构件的生产者来保证其正确性。

当该系统执行可信度量根核之外的代码时，该信任链可通过对该代码进行可信度量的方式来进行维护。如果代码的执行取决于它的可信度量，则该代码的执行授权将仍然保持不变。

以下是两种允许评估平台的信任权限的不同方法。

a) 对代码进行度量(杂凑计算)，并将度量值记录在可信存储根(RTS)中。如果代码运行时不考虑其度量值，则其信任授权来自于可信报告根(RTR)提供的摘要值。

b) 对代码进行签名，通过验证签名判断该代码是否可信。如果其身份被记录在 RTS 中，那么这个评估可被修改。

6 可信计算密码支撑平台功能

6.1 平台体系结构

6.1.1 平台功能与密码算法的关系

密码与平台功能的支撑关系见图 1。

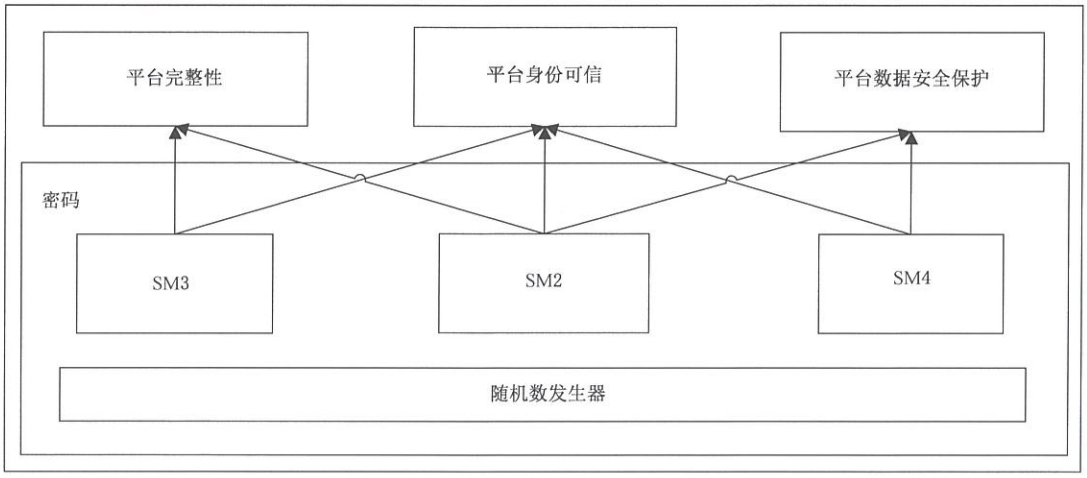

图 1 密码算法对平台功能的支撑关系图

平台的算法模块提供密码算法功能，支持和实现平台的完整性、平台身份可信及平台数据安全保护。

a) 平台完整性：利用密码机制，通过对系统平台组件的完整性度量，获取系统平台完整性信息，并向外部实体可信地报告平台完整性。

b) 平台身份可信：利用密码机制，标识系统平台身份，实现系统平台身份管理功能，并向外部实体提供系统平台身份证明。

c) 平台数据安全保护：利用密码机制，保护系统平台敏感数据。其中数据安全保护包括平台自身敏感数据的保护和用户敏感数据的保护。另外也可为用户数据保护提供服务接口。

6.1.2 平台组成结构

可信计算密码支撑平台的技术结构如图 2 所示。

注：图 2 中 TCM 服务模块和应用软件以灰色图绘制，表示非本文件重点内容。

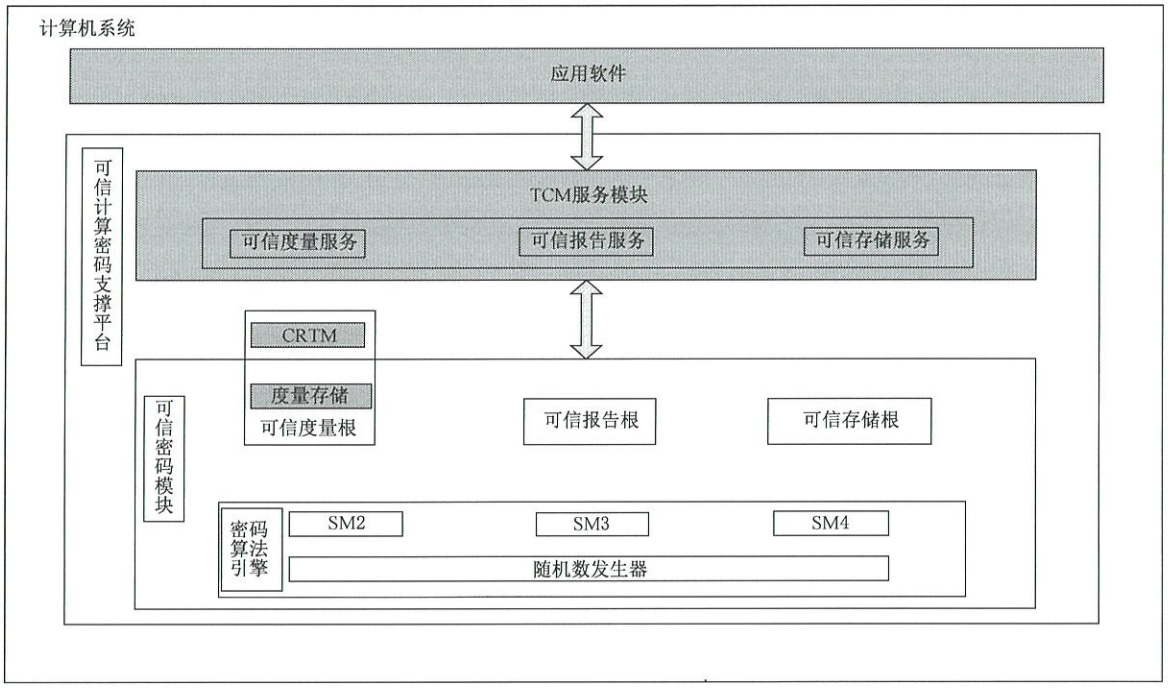

图 2 可信密码支撑平台技术结构

可信计算密码支撑平台以可信密码模块为可信根，通过如下三类机制及平台自身安全管理功能，实现平台安全功能：

a) 以可信度量根为起点，计算系统平台完整性度量值，建立计算机系统平台信任链，确保系统平台可信；

b) 可信报告根标识平台身份的可信，可信报告根基于报告根凭证派生，报告根凭证具有唯一性，以可信报告根为基础，实现平台身份证明和完整性报告；

c) 基于可信存储根，实现密钥管理、平台数据安全保护功能，提供相应的密码服务。

6.1.3 可信密码模块

可信密码模块定义了一个具有存储保护和执行保护的子系统，该子系统将为计算平台建立可信根基，并且其独立的计算资源将建立严格隔离的安全保护机制。TCM 是一个系统组件，它的状态独立于该系统并向其报告。TCM 和系统之间的唯一交互是通过本文件中定义的接口进行的。

6.1.4 TCM 服务模块

可信密码模块定义了一个具有存储保护和执行保护的子系统，该子系统将为计算平台建立信任根基，并且其独立的计算资源将建立严格隔离的安全保护机制。应将子系统中需执行保护的功能与无需执行保护的功能划分开，应由计算平台主处理器执行无需执行保护的功能，而这些支持功能构成了 TCM 服务模块。

注：本文件只说明TCM服务的功能原理，不涉及TCM服务模块接口规范，TCM服务模块接口规范见GM/T 0058。

TCM服务主要为用户使用TCM基础资源提供支持，由多个部分组成，每个部分间的接口定义应具有互操作性。TCM服务应提供规范化的函数接口。

TCM服务的设计目标如下：

a) 为应用程序调用TCM安全保护功能提供入口点；
b) 提供对TCM的同步访问；
c) 向应用程序隐藏TCM功能命令的复杂度；
d) 管理TCM资源。

6.2 平台接口功能

6.2.1 可信根

可信根是可信计算机的可信基点，也是实施安全控制的基点。可信根提供了描述平台可信度的特征所需的最小功能。

可信计算密码支撑平台中有三个可信根：可信度量根(RTM)、可信存储根(RTS)和可信报告根(RTR)。

a) RTM是对平台进行可信度量的基点。RTM将完整性信息发送给RTS。当一条新的信任链建立时，第一组被执行的指令就是可信度量根核(CRTM)。当系统复位时，CPU开始执行CRTM。然后，CRTM向RTS发送表明其身份的度量值。这样就建立了信任链的起点。
b) RTS是对可信度量值进行安全存储的基点。TCM内部存储不能被外部实体非授权访问，TCM可作为RTS。TCM可阻止对敏感信息的非授权访问，TCM也可存储一些不敏感的信息。
c) RTR是平台向访问客体提供平台可信性状态报告的基点。RTR是对于RTS内容进行报告，RTR报告是TCM内选择内容签名后的报告。

TCM包含了RTR可验证的加密标识。这个标识采用背书密钥或背书证书的形式表现，背书密钥生成的种子与每个芯片绑定，因此两个TCM不会具有相同的背书密钥。

可信计算平台至少提供上述描述RTM、RTS和RTR三个可信根。这三个根都使用认证和证明来提供信息的证据。可信计算平台还将为受托的密钥和数据对象提供受保护的存储。可信计算平台可提供完整性度量以确保平台的可信。

6.2.2 证明和认证

6.2.2.1 证明方式

6.2.2.1.1 分类

可信计算平台提供以下证明方式。

a) 第一类，外部实体对TCM进行证明，TCM应符合本文件。这种证明依据TCM中的背书密钥。
b) 第二类，外部实体对平台进行认证，以确保平台包含RTM、真实的TCM。
c) 第三类，“认证CA”对TCM中的非对称密钥对进行证明，以确保密钥受到TCM的保护，并且具有特定的自身属性。这种证明采用证书的形式，为包括密钥对的公钥在内的信息提供担保。
d) 第四类，可信计算平台对非对称密钥对进行证明，以保证密钥对受到TCM保护，并且具有特

定的自身属性。该认证方式采用在平台的 TCM 对描述密钥对的信息签名的形式，使用 TCM 保护的认证密钥，以及为该认证密钥提供担保的类型认证。

e) 第五类，可信计算平台对度量进行证明，以保证平台中存在特定的软件/固件状态。这种证明方式使用 TCM 保护的认证密钥对 PCR 中的软件/固件测量值进行签名的形式认证。

f) 第六类，外部实体对软件/固件测量进行证明。这种认证采用凭证的形式，为测量值和状态在内的信息提供签名。

6.2.2.1.2 证明密钥和证明密钥身份证书

上述第三类和第四类的认证要求使用密钥对 TCM 屏蔽位置的内容进行签名。

a) 证明密钥(AK)即平台身份密钥。AK 是一种特殊类型的签名密钥，其使用受到限制，以防止伪造(外部数据的签名具有与真实认证数据相同的格式)。当 TCM 要创建一个签名的消息时，从 TCM 内部使用一个特殊值作为消息头。当使用 AK 对摘要进行签名时，调用者提供票据，以便 TCM 应确定用于创建摘要的消息不可能是伪造的 TCM 认证数据。由 AK 签名的值可以保证反映 TCM 状态，但是 AK 也可用于一般的签名目的。

b) 证明密钥身份证书。TCM 用户可基于 TCM 创建受限使用的签名密钥，并且可要求第三方 CA 为其提供密钥身份证书。该 CA 可要求调用者提供一些证据，证明被认证的密钥是 TCM 驻留密钥，才给予提供身份证书。

6.2.2.2 平台身份标识

可信计算密码支撑平台采用背书密钥(EK)标识其身份，在 EK 授权下，在 TCM 内部生成一个非对称密钥对，作为 AK，用于对 TCM 内部的信息进行数字签名，实现平台身份认证和平台完整性报告，从而向外部证实平台内部数据的可信性。

EK 应被保存在 TCM 内，仅仅在获取背书密钥授权操作及申请平台身份证书时使用，且不得被导出到 TCM 外部。

背书密钥证书在平台使用前由一个可信方签署，用于建立 EK 与可信密码模块实例的一一对应关系。

一个可信计算密码支撑平台可产生多个 AK，每个 AK 均与 EK 绑定，对外代表平台身份。

6.2.3 受保护的位置

6.2.3.1 概述

受保护位置的保护加密使用的多个种子和密钥，这些种子和密钥永不会离开 TCM。其中之一是上下文密钥。它是一个对称密钥，用于在数据临时交换出 TCM 时加密数据，以便加载一组不同的工作对象。其他永不离开 TCM 的敏感值是主种子，这些种子是保护应用程序保留的对象的存储层次结构的根。主种子用于为其他对象生成保护密钥的随机数：这些对象可能是包含保护密钥的存储密钥。主种子可能会被更改，当它们被更改时，它们派生的对象将不再可用。

6.2.3.2 密钥管理功能

密钥管理包括以下功能。

a) 密钥生成：密钥生成是指由应用层软件设置所需生成密钥的密钥属性、密钥使用授权、密钥迁移授权、密钥的保护操作密钥，并发送给 TCM 生成指定的密钥。在 TCM 内，由保护操作密钥加密所生成的密钥私钥部分，然后将生成的密钥数据结构返回给应用层软件。对于 TCM 内各类密钥，生成方法包括：

- EK 由厂商生成，为非对称密钥；
- 平台所有者生成存储主密钥时，应在可信密码模块内部生成；
- 平台身份密钥为非对称密钥，应在 TCM 内部生成，应可信方申请相应平台身份证书并激活该密钥；
- 用户密钥可在可信密码模块内部生成，也可在可信密码模块外部生成后导入。用户密钥可以是对称密钥或非对称密钥。

b) 密钥加载：密钥生成后，在应用层软件使用该密钥进行数据安全保护操作时，如果使用该密钥的私钥，应将密钥数据加载到 TCM 内部，由保护操作密钥解密后才能使用。如果使用该密钥的公钥，则在应用层软件直接使用。可信密码模块内各类密钥的加载方法包括：
- 使用 TCM 密钥公钥在设置平台所有者之前不应验证授权，设置平台所有者之后，应验证所有者授权；密码算法运算过程应在 TCM 内部进行；
- 使用存储主密钥应验证存储主密钥授权；密码算法运算过程应在 TCM 内部进行；
- 使用平台身份密钥 AK 应验证平台身份密钥授权、存储主密钥授权；平台身份密钥的私钥应被加载到 TCM 内部进行密码算法运算，公钥的密码算法运算过程在 TCM 外部进行；
- 使用平台加密密钥应验证平台加密密钥授权、存储主密钥授权；平台加密密钥的私钥应被加载到可信密码模块内部进行密码运算，公钥的密码运算过程在模块外部进行；
- 使用用户密钥应验证用户密钥授权；用户密钥的私钥应被加载到 TCM 内部进行密码运算。

c) 密钥销毁：密钥生成后，应用层软件可销毁指定的密钥。对于 TCM 内各类密钥，销毁方法如下：
- 销毁 EK 是通过撤销可撤销的 EK 完成的，对于不可撤销的 EK，不能销毁；
- 销毁存储主密钥应验证存储主密钥授权后在 TCM 内执行；
- 销毁平台身份密钥应验证平台身份密钥授权、存储主密钥授权后，在可信密码模块内执行；
- 销毁平台加密密钥应验证平台加密密钥授权、存储主密钥授权后，在可信密码模块内执行；
- 销毁用户密钥需验证其保护操作密钥授权后，在 TCM 服务模块内执行。

d) 密钥导入：应用层软件可采用密钥导入方式，对 TCM 外部创建的非对称、对称密钥进行保护，并纳入到可信密码模块的密钥保护体系中。密钥导入方法：外部密钥需按照指定格式设定密钥属性，并使用指定的保护操作密钥对被导入密钥的私钥部分加密。

e) 密钥协商：密钥协商是在两个用户 A 和 B 之间建立一个共享秘密密钥的过程，通过这种方式确定一个共享秘密密钥的值，用来提供用户 A 与用户 B 之间传输数据的安全。

6.2.3.3 数据安全保护方式

数据加解密：数据加解密可采用对称密码算法或非对称密码算法实现。非对称加解密和对称加解密应在 TCM 内部执行。在 TCM 内部进行加解密操作时，应先加载密钥。

6.2.4 完整性度量、存储和报告

6.2.4.1 完整性度量

完整性度量以 CRTM 作为度量起点，对平台进行逐步的度量，并在 TCM 中扩展到 PCR。

完整性度量的结果是一个值，它表示平台的信任状态可能发生的变化。被度量对象通常是：

a) 数据值；

b) 代码或数据的杂凑；

c) 用户的策略配置信息或某些代码或数据的签名者的指示。

RTM(通常是运行在 CPU 上的代码)进行这些度量，并使用扩展命令在 RTS 中记录它们。该过程允许 TCM 在相对较小的内存中保存不确定数量的度量结果。用户可凭借度量结果 PCR 值和日志来确定平台的可信状态。

完整性报告是对 PCR 记录的完整性度量进行验证的过程，体现平台的完整性状态。

6.2.4.2 完整性度量与存储

完整性度量与存储是指计算组件的度量值，记录该事件到事件日志中，并把度量值扩展到可信密码模块内相应的 PCR 中。图 3 为依次度量两个组件的过程说明。

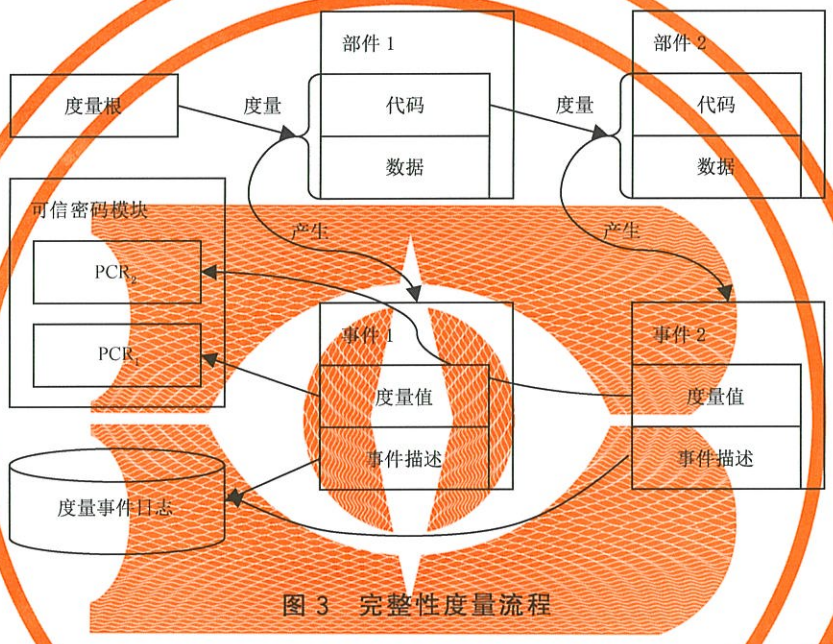

图 3 完整性度量流程

完整性度量与存储应满足如下要求。

a) 计算度量值的过程应是执行杂凑运算的过程。

b) 杂凑运算输入的数据应为度量者指定的可表征被度量者特性的数据。

c) 被度量者的完整性度量值应为杂凑运算输出的杂凑值。

d) 度量者应把度量值扩展到指定的 PCR 中。扩展的办法是：新 PCR 值＝密码杂凑算法（原 PCR 值‖度量值）。

e) 应把度量过程信息记录到平台事件日志中。至少应记录：PCR 索引、度量者信息、被度量者信息、度量值。

f) 如果一个组件序列中的各组件完整性度量值存储在同一个 PCR 中，采用如下方式：
从第一个组件开始，将该组件完整性度量值与目标 PCR 的已有存储值拼接，进行杂凑运算，然后将所得结果再存储于该 PCR 中，依次类推，最后一个组件的完整性度量值存储操作完成后，所得值即为该组件序列存储到 PCR 中的完整性度量值。

6.2.4.3 完整性报告

完整性报告是指平台向验证者提供平台或部分组件的完整性度量值的过程。完整性报告应满足如下要求：

a) 平台应向验证者提供指定 PCR 值,无需任何授权;
b) 平台应向验证者提供指定 PCR 值以及对 PCR 值的签名,签名使用平台身份密钥;
c) 平台可向验证者提供指定 PCR 的关联事件日志信息;
d) 验证者可通过分析完整性度量事件日志信息判断该 PCR 值是否来自正确的度量过程;
e) 验证者应使用平台身份密钥验证 PCR 值签名,获得平台完整性报告结果。

6.2.4.4 信任链

信任链用于保障平台完整性。平台信任链的建立是以可信度量根为起点。

首先对 BIOS 的其他组件进行完整性度量,并将度量值存储于可信密码模块的 PCR 中,按照选择的判断机制判断 BIOS 的完整性,若完整性未被破坏,则运行 BIOS;并度量初始化程序加载器(IPL)/主引导分区(MBR)的完整性,亦可基于判断机制判断 IPL/MBR 完整性,若 IPL/MBR 完整性没有被破坏,则运行 MBR;然后由 IPL/MBR 度量 OS 内核的完整性;OS 内核启动后基于同样机制检测 OS 服务完整性。通过信任关系传递,可确保所启动的系统是可信的。若上述过程中,发现某一组件的完整性受到破坏,则报告问题并按照指定策略执行操作。

7 可信密码模块接口

7.1 通用要求

规定了可信密码模块接口的具体命令,本文件所定义的可信密码模块接口应满足本章所有的命令接口,本章功能接口与 GM/T 0012 一致。本文件的所有接口涉及到的命令码、返回码和常用的数据结构应符合附录 A 的要求。

7.2 启动命令

7.2.1 TCM2_Startup

该命令用于 TCM 的初始化,当该命令执行成功之后,不再允许成功执行该命令。

当 TCM 应执行 TCM2_Startup 命令时,但收到了其他命令,或者当不应执行 TCM2_Startup 命令时收到了该命令,TCM 将返回 TCM2_RC_INITIALIZE。

以下是关闭/启动流程,定义了 TCM 收到 TCM2_Startup()命令后的操作方法。

a) TCM 重置——发送 TCM2_Shutdown(CLEAR)或者没有发送 TCM2_Shutdown()命令关闭,启动时发送 Startup(CLEAR)。当 TCM 重置时,所有的变量恢复到初始状态启动。
b) TCM 重启——发送 TCM2_Shutdown(STATE)命令关闭,启动时发送 Startup(CLEAR)。以下状态值会恢复到初始状态启动:PCR 与平台域的控制开关状态;其余 TCM 状态值将会被保存。
c) TCM 唤醒——发送 TCM2_Shutdown(STATE)命令关闭,启动时发送 TCM2_Startup(STATE)。以下状态值会以休眠保存的状态值恢复启动:
   1) S-RTM;
   2) PCR;
   3) 除 phEnable、phEnableNV 外的平台控制开关。

该命令接口输入和输出具体参数要求见表 1 和表 2。

表 1　TCM2_Startup()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	Tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Startup {NV}
TCM2_SU	startupType	TCM2_SU_CLEAR或TCM2_SU_STATE

表 2　TCM2_Startup()接口输出参数

类型	名称	描述
TCM2_ST	Tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.2.2　TCM2_Shutdown

该命令用于为下一个新的上电周期准备 TCM。shutdownType 参数用于指明下一次 TCM2_Startup()处理过程。该命令接口输入和输出具体参数要求见表 3 和表 4。

表 3　TCM2_Shutdown()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Shutdown {NV}
TCM2_SU	shutdownType	TCM2_SU_CLEAR 或 TCM2_SU_STATE

表 4　TCM2_Shutdown()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.3　检测命令

7.3.1　TCM2_SelfTest

该命令触发 TCM 执行自检功能。

当 fullTest=YES，TCM 将执行全功能自检；

当 fullTest=NO,TCM 将只对未执行自检的功能进行自检。

如果自检,TCM 将:

a) 返回 TCM2_RC_TESTING,执行需进行的功能自检;

b) 或已执行完成自检,然后返回自检成功与否。

该命令接口输入和输出参数要求见表 5 和表 6。

表 5 TCM2_SelfTest()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_SelfTest {NV}
TCMI_YES_NO	fullTest	YES:执行全功能自检 NO :对未执行自检功能执行自检

表 6 TCM2_SelfTest()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.3.2 TCM2_IncrementalSelfTest

该命令对被选中的算法执行自检。

如果 toTest 参数中包含一个已被自检过的算法,该算法将不会被再次自检。toDoList 参数中将包含仍需被自检的算法。

该命令接口输入和输出具体参数要求见表 7 和表 8。

表 7 TCM2_IncrementalSelfTest()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_IncrementalSelfTest {NV}
TCML_ALG	toTest	将被自检的算法列表

表 8 TCM2_IncrementalSelfTest()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小

表 8 TCM2_IncrementalSelfTest ( ) 接口输出参数 (续)

类型	名称	描述
TCM2_RC	responseCode	返回码
TCML_ALG	toDoList	需被测试算法列表

7.3.3 TCM2_GetTestResult

该命令返回 TCM 自检结果,outData 的结构由厂商自定义。该命令接口输入和输出具体参数要求见表 9 和表 10。

表 9 TCM2_GetTestResult()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_GetTestResult

表 10 TCM2_GetTestResult()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_MAX_BUFFER	outData	厂商自行定义
TCM2_RC	testResult	自检测试结果

7.4 会话命令

7.4.1 TCM2_StartAuthSession

该命令用于启动一个授权会话,返回会话句柄及 TCM 初始的 nonce 值。该命令提供了两种方式建立会话密钥(sessionKey),会话密钥被用于授权操作和加密参数。

该命令允许用对称或者非对称保护的方式往 TCM 中注入秘密信息。tcmKey 的类型决定了如何加密 encryptedSalt 中的值。从 encryptedSalt 中解密出来的值用于计算会话密钥(sessionKey)。

注:当 tcmKey 等于 TCM2_RH_NULL,encryptedSalt 为空值。

当 tcmKey 和 bind 都为 TCM2_ALG_NULL,会话密钥设置为空值,当 tcmKey 不为 TCM2_ALG_NULL,encryptedSalt 参数用于计算会话密钥。当 bind 不是 TCM2_ALG_NULL,bind 参数的授权值参与会话密钥的计算。

该命令接口的输入和输出具体参数要求见表 11 和表 12。

表 11 TCM2_StartAuthSession()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果当前是加密/解密会话,则为TCM2_ST_SESSIONS,否则为TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_StartAuthSession
TCMI_DH_OBJECT+	TCMKey	被加载的解密密钥句柄,salt使用该密钥进行加密 可以是TCM2_RH_NULL 授权索引:None
TCMI_DH_ENTITY+	bind	实体提供授权值 可以是TCM2_RH_NULL 授权索引:None
TCM2B_NONCE	nonceCaller	初始的nonceCaller,设置会话nonce的大小不少于16字节
TCM2B_ENCRYPTED_SECRET	encryptedSalt	使用TCMKey加密后的值 如果TCMKey是TCM2_RH_NULL,该值将为空
TCM2_SE	sessionType	指明会话的类型,简单的HMAC或策略(包括试用策略)
TCMT_SYM_DEF+	symmetric	参数加密的算法及密钥长度,可为TCM2_ALG_NULL
TCMI_ALG_HASH	authHash	会话使用的SM3杂凑算法,不能是TCM2_ALG_NULL

表 12 TCM2_StartAuthSession()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag应是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMI_SH_AUTH_SESSION	sessionHandle	新创建的会话句柄
TCM2B_NONCE	nonceTCM	来自TCM的初始nonce值,用于计算会话密钥sessionKey

7.4.2 TCM2_PolicyRestart

该命令使增强授权会话返回到初始状态。当TCM返回TCM2_RC_PCR_CHANGED时该命令将会被使用。

该命令不会重置policyID或者策略开始的时间。

该命令接口输入和输出具体参数要求见表13和表14。

表 13 TCM2_PolicyRestart()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyRestart
TCMI_SH_POLICY	sessionHandle	策略会话句柄

表 14 TCM2_PolicyRestart()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.5 对象命令

7.5.1 TCM2_Create

该命令用于创建一个可被 TCM2_Load 命令载入 TCM 内部的对象，如果命令执行成功，TCM 将创建新对象，并返回对象的创建数据(creationData)、公开部分(outPublic)和加密敏感区域(outPrivate)，返回的数据由调用者负责保存。该对象使用之前应调用 TCM2_Load 命令载入 TCM。

该命令接口输入和输出具体参数要求见表 15 和表 16。

表 15 TCM2_Create()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Create
TCMI_DH_OBJECT	@parentHandle	新对象的父句柄，父句柄应指向一个已载入 TCM 的解密密钥 授权索引：1 授权角色：USER
TCM2B_SENSITIVE_CREATE	inSensitive	隐私数据
TCM2B_PUBLIC	inPublic	公开数据
TCM2B_DATA	outsideInfo	包含在此对象的创建数据中的数据，用于提供对象与对象拥有者数据之间永久可认证的联系
TCML_PCR_SELECTION	creationPCR	指定在创建数据(creationData)中使用的 PCR 寄存器

表 16 TCM2_Create()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag应是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_PRIVATE	outPrivate	对象的隐私数据部分
TCM2B_PUBLIC	outPublic	对象的公开数据部分
TCM2B_CREATION_DATA	creationData	创建数据,包含一个TCMS_CREATION_DATA结构
TCM2B_DIGEST	creationHash	使用公开数据中的杂凑算法(nameAlg)对creationData计算的摘要值
TCMT_TK_CREATION	creationTicket	TCM2_CertifyCreation()命令使用的票据,用来验证由TCM产生的创建数据(creationData)

7.5.2 TCM2_Load

该命令用于将一个对象载入 TCM 中,只有当对象的公开部分(TCM2B_PUBLIC)和隐私部分(TCM2B_PRIVATE)都被载入 TCM 时此命令才被使用。

如果只有公开部分被载入 TCM,TCM2_LoadExternal 命令将被使用。

如果只有部分公开部分被载入 TCM,TCM2_LoadExternal 命令将被使用。

此命令将返回被载入对象的句柄和名字,名字是由 TCM 计算的 inPublic 结构的 TCMT_PUBLIC 成员的摘要值。

如果对象的隐私数据部分 inPrivate.size 为 0,加载将失败。

该命令接口输入和输出具体参数要求见表 17 和表 18。

表 17 TCM2_Load()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Load
TCMI_DH_OBJECT	@parentHandle	父密钥的句柄,不应是一个被保留的句柄 授权索引:1 授权角色:USER
TCM2B_PRIVATE	inPrivate	对象的隐私数据部分
TCM2B_PUBLIC	inPublic	对象的公开数据部分

表 18 TCM2_Load()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag应是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2_HANDLE	objectHandle	载入对象的句柄
TCM2B_NAME	name	载入对象的名字

7.5.3 TCM2_LoadExternal

该命令用于将一个非保护性的对象载入 TCM,允许载入对象的公开部分,用于验证签名,或者既载入对象的公开部分又载入隐私部分,用于加密操作。此命令将返回载入对象的句柄和名字,名字是摘要算法 nameAlg 和对象公开数据部分摘要值的级联。

当只载入一个外部对象的公开部分时,允许制定一个对象关联的层级,这样创建票据时可使用正确的算法,hierarchy 参数指定了这个层级,如果对象的公开部分和隐私部分都被载入,hierarchy 参数被要求是 TCM2_RH_NULL。

该命令接口输入和输出具体参数要求见表 19 和表 20。

表 19 TCM2_LoadExternal()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密/解密操作,tag为TCM2_ST_SESSIONS,否则为TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_LoadExternal
TCM2B_SENSITIVE	inPrivate	对象的隐私数据部分(可选)
TCM2B_PUBLIC+	inPublic	对象的公开数据部分
TCMI_RH_HIERARCHY+	hierarchy	对象关联的层级

表 20 TCM2_LoadExternal()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2_HANDLE	objectHandle	被载入对象的句柄
TCM2B_NAME	name	被载入对象的名字

7.5.4 TCM2_ReadPublic

该命令返回一个已载入对象的公开数据部分，该对象句柄的使用无需授权信息。如果 objectHandle 引用序列对象，则 TCM 应返回 TCM2_RC_sequence。该命令接口输入和输出具体参数要求见表 21 和表 22。

表 21 TCM2_ReadPublic()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密会话，tag 为 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ReadPublic
TCMI_DH_OBJECT	objectHandle	对象的句柄 授权索引：None

表 22 TCM2_ReadPublic()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_PUBLIC	outPublic	包含对象的公开数据的结构体
TCM2B_NAME	name	对象的名字
TCM2B_NAME	qualifiedName	对象的限定名称

7.5.5 TCM2_ActivateCredential

该命令给指定的对象关联一个凭证，确保 TCM 已经认证了该对象的参数。

如果参数 activateHandle 和 KeyHandle 的公开部分和隐私部分都没有被加载，则 TCM 返回 TCM2_RC_AUTH_UNAVAILABLE。

KeyHandle 应是一个存储密钥，否则 TCM 返回 TCM2_RC_TYPE。

对 ActivateHandle 的授权要求管理员身份。

与 keyHandle 关联的密钥用于从 secret 恢复种子，即加密种子。

与 activateHandle 关联的对象的名称和恢复的种子用于恢复对称密钥，恢复的种子（而不是名称）用于恢复 hmac 密钥。

HMAC 被用来认证输入参数 credentialBlob 与 activateHandle 关联，且 credentialBlob 中的数据没有被改变。

如果 credentialBlob 数据的完整性检查成功，credentialBlob 被解密，作为输出参数 certInfo 返回。

该命令接口输入和输出具体参数要求见表 23 和表 24。

表 23 TCM2_ActivateCredential()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ActivateCredential
TCMI_DH_OBJECT	@activateHandle	与 credentialBlob 中的证书关联的对象句柄 授权索引：1 授权身份：ADMIN
TCMI_DH_OBJECT	@keyHandle	加载的密钥，用来解密在 credentialBlob 中的 TCMS_SENSITIVE 授权索引：2 授权身份：USER
TCM2B_ID_OBJECT	credentialBlob	对象凭证
TCM2B_ENCRYPTED_SECRET	secret	依赖于密钥算法的种子，用来保护凭证信息

表 24 TCM2_ActivateCredential()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	certInfo	解密后的证书信息 该数据长度不应大于 keyHandle 密钥关联的 nameAlg 指定的摘要算法产生的摘要长度

7.5.6 TCM2_MakeCredential

该命令允许 TCM 执行证书颁发机构在创建包含激活凭据的 TCM2B_ID_OBJECT 时所需的操作。输入参数 handle 的公开区域要求是存储密钥类型，否则凭证不能被正确的绑定。

该命令接口输入和输出具体参数要求见表 25 和表 26。

表 25 TCM2_MakeCredential()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密/解密会话，tag 为 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_MakeCredential

表 25 TCM2_MakeCredential( ) 接口输入参数 (续)

类型	名称	描述
TCMI_DH_OBJECT	handle	handle指向的对象的载入公开区域,用来加密包含凭证密钥的敏感数据 认证索引:None
TCM2B_DIGEST	credential	凭证信息
TCM2B_NAME	objectName	此凭证关联的对象名称

表 26 TCM2_MakeCredential()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ID_OBJECT	credentialBlob	凭证信息
TCM2B_ENCRYPTED_SECRET	secret	包含用来加密凭证信息的密钥的秘密数据

7.5.7 TCM2_Unseal

该命令返回一个被载入 TCM 的封装数据对象的关联数据。这个封装数据对象可能是 TCM 自己产生的,但更有可能是外部创建,通过 TCM2_Import 导入 TCM 的。该命令的返回值可能通过授权会话加密。

如果封装数据对象属性中的 restricted, decrypt 或 sign 比特被设置为 1,则 TCM 返回 TCM2_RC_ATTIBUTES。如果 itemHandle 类型不是 TCM2_ALG_KEYEDHASH,则 TCM 返回 TCM2_RC_TYPE。

该命令接口输入和输出具体参数要求见表 27 和表 28。

表 27 TCM2_Unseal()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	Tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Unseal
TCMI_DH_OBJECT	@itemHandle	被加载的封装数据对象的句柄 授权索引:1 授权身份:USER

表 28 TCM2_Unseal()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_SENSITIVE_DATA	outData	解封装得到的数据 解封数据的大小不能超过128字节

7.5.8 TCM2_ObjectChangeAuth

该命令用来改变已载入 TCM 对象的授权信息。如果命令执行成功,将返回该对象新的授权值,此命令不改变该对象操作的其他已载入 TCM 的对象的授权值。此命令不能改变一个 NV 索引对象的授权值。

如果 NV 索引要有新的授权,则使用 TCM2_NV_ChangeAuth()完成。

如果要改变一个主要对象的授权值,调用 TCM2_CreatePrimary 命令重新创建该对象。

该命令接口输入和输出具体参数要求见表 29 和表 30。

表 29 TCM2_ObjectChangeAuth()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ObjectChangeAuth
TCMI_DH_OBJECT	@objectHandle	对象句柄 授权索引:1 授权身份:ADMIN
TCMI_DH_OBJECT	parentHandle	父密钥句柄 授权索引:None
TCM2B_AUTH	newAuth	新的授权信息

表 30 TCM2_ObjectChangeAuth()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_PRIVATE	outPrivate	包含新授权信息的对象隐私数据部分

7.6 复制命令

7.6.1 TCM2_Duplicate

该命令复制一个已加载的对象使得该对象可被用在不同的层次上。新的父密钥可能在相同的TCM中,也可能在不同的TCM中,也可能是TCM2_RH_NULL。该命令只要求新的父密钥的公开部分被加载。

该命令应有一个策略会话的授权。如果TCM2_PolicyCpHash作为策略的一部分已经被执行,则策略会话的cpHash值应与命令的cpHash值相等,否则TCM返回TCM2_RC_POLICY_FAIL。

该命令接口输入和输出具体参数要求见表31和表32。

表31 TCM2_Duplicate()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Duplicate
TCMI_DH_OBJECT	@objectHandle	将被复制的已加载对象 授权索引:1 授权身份:DUP
TCMI_DH_OBJECT+	newParentHandle	父密钥的公开区域 授权索引:None
TCM2B_DATA	encryptionKeyIn	可选的对称加密密钥 如果该密钥由TCM生成,密钥长度被设置为0, 该参数可能是被加密的
TCMT_SYM_DEF_OBJECT+	symmetricAlg	内部加密使用的对称加密算法 如果不存在内部加密,则该参数应是TCM2_ALG_NULL

表32 TCM2_Duplicate()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DATA	encryptionKeyOut	如果调用者提供一个加密密钥或者加密算法是TCM2_ALG_NULL,该参数将返回Empty Buffer,否则返回一个TCM生成的用于内部加密的对称密钥
TCM2B_PRIVATE	duplicate	可通过encryptionKeyIn加密的私有区域;可能是双重加密的
TCM2B_ENCRYPTED_SECRET	outSymSeed	新父密钥的种子,由新父密钥加密保护

7.6.2 TCM2_Rewrap

该命令允许 TCM 履行一个迁移授权的角色。如果使用父密钥的授权信息被提供，则可从参数 inSymKey 中获得一个对称密钥，用来进行完整性检查和解密迁移对象的私钥部分。根据新的父密钥算法，一个新的保护种子被生成，迁移对象的私钥部分被重新加密，并产生一个新的完整性值。重新加密后的私钥部分和新的对称密钥在参数 outDuplicate 和 outSymKey 中返回。

该命令接口输入和输出具体参数要求见表 33 和表 34。

表 33 TCM2_Rewrap()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Rewrap
TCMI_DH_OBJECT+	@oldParent	对象的父密钥句柄 授权索引：1 授权身份：User
TCMI_DH_OBJECT+	newParent	对象的新父密钥句柄 授权索引：None
TCM2B_PRIVATE	inDuplicate	inSymSeed生成的对称密钥加密后的隐私部分
TCM2B_NAME	name	被迁移的对象名字
TCM2B_ENCRYPTED_SECRET	inSymSeed	对称密钥的种子使用旧父密钥的私钥部分恢复种子以产生对称密钥

表 34 TCM2_Rewrap()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_PRIVATE	outDuplicate	新种子生成的对称密钥加密后的迁移密钥隐私部分
TCM2B_ENCRYPTED_SECRET	outSymSeed	由新父密钥加密过的对称密钥的种子

7.6.3 TCM2_Import

该命令允许一个对象被一个存储密钥的对称加密值加密。加密之后，该对象可被载入，并在一个新的等级上使用。被导入的对象可能被一次或多次加密，也可能没有加密。该命令接口输入和输出具体参数要求见表 35 和表 36。

被导入对象隐私部分的恢复分为三步。

a) 如果外层对称加密存在，则进行外层解密：用于产生 HMAC 密钥和加密密钥的种子用父密钥的私钥恢复，这些密钥之后被用来保护被导入数据的隐私部分。

b) 如果内层加密存在，则进行内层解密。使用 encryptionKey 和 symmetricAlg 指定的加密算法解密被导入数据的隐私部分。

c) 如果隐私数据的完整性值存在，则进行完整性检查，完整性值用导入数据的公开部分的名字进行验证。

解密操作和完整性检查完成之后，TCM 将创建一个新的被父密钥的加密密钥加密过的隐私区域。该命令接口输入和输出具体参数要求见表 35 和表 36。

表 35 TCM2_Import()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Import
TCMI_DH_OBJECT	@parentHandle	对象的父密钥句柄 授权索引：1 授权身份：USER
TCM2B_DATA	encryptionKey	可选的用于内部加密的对称加密密钥 如果加密算法参数为 TCM2_ALG_NULL，那么该参数应为 Empty Buffer
TCM2B_PUBLIC	objectPublic	被导入对象的公开区域 为了检查 duplicate 和 object 属性的完整性
TCM2B_PRIVATE	duplicate	对称加密的 duplicate 对象 可能包含内部对称封装
TCM2B_ENCRYPTED_SECRET	inSymSeed	用于加密 duplicate 的对称密钥 使用父密钥的算法加密/编码
TCMT_SYM_DEF_OBJECT+	symmetricAlg	指定用来进行内层加密的加密算法 如果该参数为 TCM2_ALG_NULL，表明不存在内层加密，且 encryptionKey 应为 Empty Buffer

表 36 TCM2_Import()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_PRIVATE	outPrivate	用 parentHandle 的对称密钥加密的敏感区域

7.7 非对称算法命令

7.7.1 TCM2_ZGen_2Phase

该命令用于 SM2 密钥的密钥协商操作。

SM2 密钥协商有两个阶段，该命令与 TCM2_EC_Ephemal()命令配合使用。SM2 密钥交换协议应符合 GB/T 32918.3，使用规范应符合 GB/T 35276。

TCM2_EC_Ephemeral()命令用于生成短暂生命周期的密钥(短暂密钥)并返回该密钥的公钥及一些数据值，使用这些数据值 TCM 可重新生成该密钥对应的私钥部分。

该命令的输入参数为来自 B 方的静态公钥(inQsU)，短暂密钥(inQeU)，以及 TCM2_EC_Ephemeral()命令返回的参数 commitCounter.TCM 使用计数器值重新计算短暂私钥 ($d_{e,V}$)以及对应的公钥($Q_{e,V}$)，keyA 提供静态短暂元素($d_{s,V}$)和($Q_{s,V}$)，TCM 将计算出 Z_s 和 Z_e 的值。

该命令接口输入和输出具体参数要求见表 37 和表 38。

表 37　TCM2_ZGen_2Phase()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ZGen_2Phase
TCMI_DH_OBJECT	@keyA	非限制 ECC 解密密钥句柄 该句柄的私钥为 $d_{s,A}$ 授权索引：1 授权角色：USER
TCM2B_ECC_POINT	inQsB	第三方静态密钥公钥($Q_{s,B}=(X_{s,B}, Y_{s,B})$)
TCM2B_ECC_POINT	inQeB	第三方短暂密钥公钥($Q_{e,B}=(X_{e,B}, Y_{e,B})$)
TCMI_ECC_KEY_EXCHANGE	inScheme	密钥交换策略
UINT16	counter	TCM2_EC_Ephemeral()返回的值

表 38　TCM2_ZGen_2Phase()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	outZ1	计算值的 X 轴和 Y 轴坐标
TCM2B_ECC_POINT	outZ2	第二次计算值得 X 轴和 Y 轴坐标

7.7.2　TCM2_ECC_Encrypt

该命令用于完成 SM2 密钥的非对称加密操作。keyHandle 参数指向用于非对称加密操作的密钥。根据加密方案，返回临时协商公钥(C1)、加密数据(C2)和杂凑值(C3)。其中，SM2 加密算法应符合 GB/T 32918.4。

该命令接口输入和输出具体参数要求见表 39 和表 40。

表 39 TCM2_ECC_Encrypt()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ECC_Encrypt
TCMI_DH_OBJECT	keyHandle	用于执行加密密钥的公钥部分 授权索引：None
TCM2B_MAX_BUFFER	plaintext	待加密数据
TCMT_KDF_SCHEME+	inScheme	如果与keyHandle关联的scheme为TCM2_ALG_NULL，则要使用KDF

表 40 TCM2_ECC_Encrypt()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	C1	临时协商公钥
TCM2B_MAX_BUFFER	C2	加密数据，通过XOR处理
TCM2B_DIGEST	C3	杂凑值

7.7.3 TCM2_ECC_Decrypt

该命令用于完成SM2密钥的非对称解密操作。keyHandle参数指向用于非对称解密操作的密钥。该命令接口输入和输出具体参数要求见表41和表42。

表 41 TCM2_ECC_Decrypt()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ECC_Decrypt
TCMI_DH_OBJECT	@keyHandle	用于执行解密密钥的公钥部分 授权索引：1
TCM2B_ECC_POINT	C1	临时协商公钥
TCM2B_MAX_BUFFER	C2	加密数据，通过XOR处理
TCM2B_DIGEST	C3	杂凑值
TCMT_KDF_SCHEME+	inScheme	如果与keyHandle关联的scheme为TCM2_ALG_NULL，则要使用KDF

表 42 TCM2_ECC_Decrypt()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag应是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_MAX_BUFFER	plainText	已解密的数据输出

7.7.4 TCM2_ECDH_ZGen

该命令使用 TCM 从一个公共的点(Q_B)公钥部分和私钥(d_s)来恢复 Z 的值。它将执行提供的 inPoint(Q_B)与私钥(d_s)的乘法,并返回 point[$Z=(x_Z, y_Z):=[hd_s]Q_B$;其中 h 是曲线的余因子]坐标的结果。

keyHandle 为已加载的受限属性 CLEAR 和解密属性集(TCM2_RC_ATTRIBUTES)的 ECC 密钥(TCM2_RC_KEY)。

注:当属性为TCM2_RC_ATTRIBUTES时,TCM2_RC_KEY是可接受的。

被keyHandle引用的密钥方案应是TCM2_ALG_ECDH或TCM2_ALG_NULL(TCM2_RC_scheme)。

inPoint 应位于 keyHandle 引用的密钥的曲线上(TCM2_RC_ECC_POINT)。

被 keyHandle 引用的密钥参数用于执行点乘。

该命令接口输入和输出具体参数要求见表 43 和表 44。

表 43 TCM2_ECDH_ZGen()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ECDH_ZGen
TCMI_DH_OBJECT	@keyHandle	加载的ECC密钥句柄 授权索引:1 授权角色:USER
TCM2B_ECC_POINT	inPoint	公钥

表 44 TCM2_ECDH_ZGen()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	outPoint	X与Y坐标乘积,$Z=(x_Z, y_Z):=[hd_S]Q_B$

7.7.5 TCM2_ECDH_KeyGen

这个命令使用 TCM 生成一个临时密钥对(d_e,Q_e,其中 Q_e:=[d_e]G),它使用私有临时密钥和加载的公钥(Q_S)来计算共享密钥值(P:=[hd_e]Q_S)。

KeyHandle 是加载的 ECC 密钥(TCM 密钥)。keyHandle 不应加载此密钥的敏感部分。利用加载的 ECC 密钥的曲线参数生成临时密钥。

注:此函数相当于将数据加密到另一个对象的公钥。种子的值在 KDF 中用于生成对称密钥,该密钥用于加密数据。一旦数据被加密并且对称密钥被丢弃,只有具有密钥句柄私有部分的对象才能对其进行解密。

zPoint 的返回值可使用参数加密进行加密。

该命令接口输入和输出参数具体参数要求见表 45 和表 46。

表 45 TCM2_ECDH_KeyGen()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密会话, tag=TCM2_ST_SESSIONS, 否则,tag=TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ECDH_KeyGen
TCMI_DH_OBJECT	@keyHandle	已加载的 ECC 公钥的句柄 授权索引:None

表 46 TCM2_ECDH_KeyGen()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	zPoint	P:=h[d_e]Q_s 的结果
TCM2B_ECC_POINT	pubPoint	产生临时的公共点 point(Q_e)

7.8 对称算法命令

TCM2_EncryptDecrypt 命令用于执行对称加解密操作。SM4 对称密码算法应符合 GB/T 32907。

keyHandle 参数指向对称密钥对象(TCM2_RC_KEY)。对于受限密钥,mode 参数应与密钥的模式相同,或为 TCM2_ALG_NULL;对于非受限密钥,mode 参数可与密钥的模式相同或者不同但是不能同为 TCM2_ALG_NULL(TCM2_RC_VALUE)。

该命令接口输入和输出具体参数见表 47 和表 48。

表 47　TCM2_EncryptDecrypt()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_EncryptDecrypt
TCMI_DH_OBJECT	@keyHandle	用于操作的对称密钥 授权索引:1 授权角色:USER
TCM2B_MAX_BUFFER	inData	待加密/解密的数据
TCMI_YES_NO	decrypt	如果是YES,当前为解密操作 如果是NO,当前为加密操作
TCMI_ALG_CIPHER_MODE+	mode	对称模式 该字段应与密钥的默认模式匹配或为TCM2_ALG_NULL
TCM2B_IV	ivIn	算法要求的初始值

表 48　TCM2_Encryp.tDecrypt()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_MAX_BUFFER	outData	加密或者解密输出
TCM2B_IV	ivOut	下一循环使用的IV值

7.9　随机数发生器命令

7.9.1　TCM2_GetRandom

该命令将返回RNG发生器中的下一个bytesRequested字节长度的随机数。当bytesRequested的长度大于TCM2B_DIGEST结构中的摘要长度,TCM不会发生错误,返回的随机数的长度与TCM2B_DIGEST中摘要长度一致,产生的随机数应符合GB/T 32915。

该命令接口输入和输出具体参数要求见表49和表50。

表 49 TCM2_GetRandom()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密会话,则是TCM2_ST_SESSIONS,否则为TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_GetRandom
UINT16	bytesRequested	需要生成的随机数的长度

表 50 TCM2_GetRandom()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出标记为TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	randomBytes	随机数

7.10 杂凑/HMAC命令

7.10.1 TCM2_Hash

该命令基于数据缓存区执行杂凑运算SM3并返回运算结果。SM3杂凑算法应符合GB/T 32905。

注:如果数据缓存区中待计算的数据长度大于TCM的数据输入缓存区,使用杂凑序列命令。

当杂凑运算的结果在被使用受限签名密钥进行签名操作时,该命令返回的票据信息表明该杂凑值是安全的,可被签名。

当返回的杂凑结果不安全时,TCM将返回TCMT_TK_HASHCHECK,hierarchy设置为TCM2_RH_NULL,digest设置为空。

当hierarchy等于TCM2_RH_NULL,票据中的摘要digest设置为空。

该命令接口输入和输出具体参数要求见表51和表52。

表 51 TCM2_Hash()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密/解密会话,则是TCM2_ST_SESSIONS,否则为TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Hash
TCM2B_MAX_BUFFER	data	参与杂凑运算的数据
TCMI_ALG_HASH	hashAlg	杂凑运算中用到的算法:不能为TCM2_ALG_NULL
TCMI_RH_HIERARCHY+	hierarchy	分层使用的票据(允许为TCM2_RH_NULL)

表 52 TCM2_Hash()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出标记应是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	outHash	杂凑运算结果
TCMT_TK_HASHCHECK	validation	用于指明被用于计算 outDigest 的数据序列不是以 TCM2_GENERATED_VALUE 开头,如果该摘要不使用受限密钥进行签名,该值可为空

7.10.2 TCM2_HMAC

该命令使用提供的数据集指明的 SM3 杂凑算法执行 HMAC 运算。

执行者使用 handle 时应提供恰当的授权信息。

当 handle 密钥的 handle 指向的密钥的签名属性没有设置,TCM 将返回 TCM2_RC_ATTRIBUTES. 如果密钥的类型不是 TCM2_ALG_KEYEDHASH TCM 将返回 TCM2_RC_TYPE。

该命令接口输入和输出具体参数要求见表 53 和表 54。

表 53 TCM2_HMAC()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_HMAC
TCMI_DH_OBJECT	@handle	HAMC 计算使用的对称密钥句柄授权 索引:1 授权角色:USER
TCM2B_MAX_BUFFER	buffer	HMAC 数据
TCMI_ALG_HASH+	hashAlg	HMAC 使用的杂凑算法

表 54 TCM2_HMAC()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	outHMAC	返回的 HMAC 计算值

7.10.3 TCM2_HMAC_Start

该命令启动一个 HMAC 序列，TCM 将会创建并初始化一个 HMAC 序列结构，分配一个句柄给该序列，并设置序列对象的授权值为 auth。

调用者使用 handle 时应提供恰当的授权值。当 handle 指向的密钥的签名属性没有设置时，TCM 将返回 TCM2_RC_ATTRIBUTES，如果密钥的类型不是 TCM2_ALG_KEYEDHASH，TCM 将返回 TCM2_RC_TYPE。

该命令接口输入和输出具体参数要求见表 55 和表 56。

表 55 TCM2_HMAC_Start()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_HMAC_Start
TCMI_DH_OBJECT	@handle	HMAC密钥句柄 授权索引：1 授权角色：USER
TCM2B_AUTH	auth	子序列使用该序列的授权值
TCMI_ALG_HASH+	hashAlg	该HMAC使用的杂凑算法

表 56 TCM2_HMAC_Start()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMI_DH_OBJECT	sequenceHandle	指向该序列的句柄

7.10.4 TCM2_HashSequenceStart

该命令启动一个杂凑或者事件序列。

如果 hashAlg 是一个可被执行的 SM3 杂凑算法，杂凑序列将被执行。如果 hashAlg 是 TCM2_ALG_NULL，将会启动一个事件序列。如果 hashAlg 即不是可执行杂凑算法又不是 TCM2_ALG_NULL，TCM 将会返回 TCM2_RC_HASH。

根据 hashAlg 的不同，TCM 将会创建和初始化杂凑序列结构或者事件序列结构。例如，TCM 将会分配一个具备给序列并设置序列的授权值为 auth。一个事件序列的结构包含 TCM 中每个 PCR 执行的上下文信息。

该命令接口输入和输出具体参数要求见表 57 和表 58。

表 57　TCM2_HashSequenceStart()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_HashSequenceStart
TCM2B_AUTH	auth	子序列使用该序列的授权值
TCMI_ALG_HASH+	hashAlg	杂凑序列用的杂凑算法，如果为 TCM2_ALG_NULL 则为事件序列

表 58　TCM2_HashSequenceStart()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMI_DH_OBJECT	sequenceHandle	指向序列的句柄

7.10.5　TCM2_SequenceUpdate

该命令被用于给杂凑或者 HMAC 序列增加数据。Buffer 中的数据可以是小于 TCM 允许值的任何值。

sequenceHandle 参数关联的序列对象应使用恰当的授权。

如果与该命令关联的授权要计算 cpHash 和 rpHash 值，与 sequenceHandle 关联的 Name 参数将会为空。

该命令接口输入和输出具体参数要求见表 59 和表 60。

表 59　TCM2_SequenceUpdate()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_SequenceUpdate
TCMI_DH_OBJECT	@sequenceHandle	指向序列对象的句柄 授权索引：1 授权角色：USER
TCM2B_MAX_BUFFER	buffer	参加杂凑计算的数据

表 60 TCM2_SequenceUpdate()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.10.6 TCM2_SequenceComplete

该命令增加杂凑/HMAC序列数据的最后部分,并返回计算结果。

对于杂凑序列,如果该命令计算结果将被用于受限密钥执行的签名操作,那么该命令返回的票据信息可表明该杂凑值是安全的,可用于签名操作。

如果digest不是安全的,validation将会为TCMT_TK_HASHCHECK,该结构中hierarchy值为TCM2_RH_NULL,digest值为空。

当sequenceHandle指向事件序列,TCM将返回TCM2_RC_MODE。

sequenceHandle参数关联的序列对象应使用恰当的授权。如果与该命令关联的授权计算cpHash和rpHash值,与sequenceHandle关联的Name参数将会为空。

当命令成功执行,sequenceHandle对象将被释放。

该命令接口输入和输出具体参数要求见表61和表62。

表 61 TCM2_SequenceComplete()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_SequenceComplete
TCMI_DH_OBJECT	@sequenceHandle	序列使用的授权索引:1 授权角色:USER
TCM2B_MAX_BUFFER	buffer	参加杂凑/HMAC运算的数据
TCMI_RH_HIERARCHY+	hierarchy	杂凑操作的分层票据

表 62 TCM2_SequenceComplete()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	result	HMAC返回值或是缓存区的摘要值
TCMT_TK_HASHCHECK	validation	票据表明参与outDigest计算的数据序列不是以TCM2_GENERATED_VALUE开头,当为HMAC会话时,该值为NULL

7.11 证明命令

7.11.1 TCM2_Certify

该命令的目的是证明某特定命名的对象被加载到 TCM 中。通过证明被加载的对象，TCM 保证一个公共区域与一个给定名字是一致，且与一个有效敏感区联系。当依赖方有一个公共区，该公共区的名字与使用该命令进行了认证的名字相同，那么他的公共区的值也是正确的。

本文件证书格式应符合 GB/T 20518。

objectHandle 的授权应为 ADMIN 角色的授权值。如果使用了策略会话，那么 policySession→commandCode 设置为 TCM2_CC_Certify。这表明该增强会话认证使用，不能用于其他用途。

objectHandle 可为 TCM2_Load()或 TCM2_CreatePrimary()任意对象。不能对只加载公共区域的对象。

该命令接口输入和输出具体参数要求见表 63 和表 64。

表 63 TCM2_Certify()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Certify
TCMI_DH_OBJECT	@objectHandle	被证明对象的句柄 授权索引：1 授权角色：ADMIN
TCMI_DH_OBJECT+	@signHandle	用于签名证明结构的密钥句柄 授权索引：2 授权角色：USER
TCM2B_DATA	qualifyingData	用户提供的资格数据
TCMT_SIG_SCHEME+	inScheme	如果 sigHanlde 的签名策略为 TCM2_ALG_NULL，使用该数据作为签名策略

表 64 TCM2_Certify()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ATTEST	certifyInfo	签名后的结构
TCMT_SIGNATURE	signature	使用 signHandle 密钥对 certifyInfo 进行非对称签名后的结构

7.11.2 TCM2_CertifyCreation

该命令用于为一个对象及其产生的数据提供证明。TCM 将会验证票据是由 TCM 生成，并且该票据将验证加载的公共区域和提供的 creation 数据的杂凑值(creationHash)之间的关联。

TCM 将使用对象名、objectHandle 及 creationHash 计算一个测试票据。

HMAC(proof, (TCM2_ST_CREATION ‖ objectHandle → Name ‖ CreationHash))

TCM 将使用生成的票据与输入时提供的票据进行比较，如果不同，TCM 返回 TCM2_RC_TICKET。

如果输入的票据是有效的，TCM 将创建一个 TCMS_ATTEST 结构，把 creationHash 的值填充在该结构的 creationHash 域中。

名称以及 objectHandle 将被包含在证明数据中，使用 signHandle 参数指向的密钥对该证明数据进行签名操作。

ObjectHandle 可为 TCM2_Load()或者 TCM2_CreatePrimary()加载的任意对象。

该命令接口输入和输出具体参数要求见表 65 和表 66。

表 65 TCM2_CertifyCreation()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_CertifyCreation
TCMI_DH_OBJECT+	@signHandle	对证明数据块进行签名的密钥句柄 授权索引：1 授权角色：USER
TCMI_DH_OBJECT	objectHandle	与产生的数据关联的对象 授权索引：None
TCM2B_DATA	qualifyingData	用户提供的资格数据
TCM2B_DIGEST	creationHash	TCM2_Create()或 TCM2_CreatePrimary()命令生成的杂凑值
TCMT_SIG_SCHEME+	inScheme	如果 signHandle 密钥的签名策略为 TCM2_ALG_NULL，使用该值作为签名策略
TCMT_TK_CREATION	creationTicket	TCM2_Create()或 TCM2_CreatePrimary()创建的票据
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS

表 66 TCM2_CertifyCreation()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ATTEST	certifyInfo	被签名的结构
TCMT_SIGNATURE	signature	基于certifyInfo的签名值

7.11.3 TCM2_Quote

该命令被用于引用 PCR 值。

TCM 将对 PCRSelect 参数选择的 PCR 并使用 signHandle 指向的密钥关联的杂凑算法进行杂凑运算。

该命令接口输入和输出具体参数要求见表 67 和表 68。

表 67 TCM2_Quote()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Quote
TCMI_DH_OBJECT	@signHandle	执行签名操作的密钥句柄 授权索引:1 授权角色:USER
TCM2B_DATA	qualifyingData	调用者提供的资格数据
TCMT_SIG_SCHEME+	inScheme	signHanle为TCM2_ALG_NULL的使用的签名方式,如果signHandle的签名策略为TCM2_ALG_NULL,使用该值作为签名策略
TCM_PCR_SELECTION	PCRSelect	引用的PCR

表 68 TCM2_Quote()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ATTEST	quoted	被引用的信息
TCMT_SIGNATURE	signature	基于quoted的签名值

7.11.4 TCM2_GetTime

该命令表示返回当前的时间。该命令接口输入和输出具体参数要求见表 69 和表 70。

表 69 TCM2_GetTime()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_GetTime
TCMI_RH_ENDORSEMENT	@privacyAdminHandle	私有所有者句柄(TCM2_RH_ENDORSEMENT) 授权索引：1 授权角色：USER
TCMI_DH_OBJECT+	@signHandle	能执行签名操作的密钥句柄 授权索引：2 授权角色：USER
TCM2B_DATA	qualifyingData	资格数据
TCMT_SIG_SCHEME+	inScheme	如果 signHandle 中的签名策略为 TCM2_ALG_NULL，使用该值作为签名策略

表 70 TCM2_GetTime()接口输出参数

类型	名称	描述
TCM2_ST	Tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ATTEST	timeInfo	标准 TCM 创建的证明
TCMT_SIGNATURE	signature	基于 timeInfo 的签名

7.12 临时EC密钥命令

7.12.1 TCM2_EC_Ephemeral

该命令用于为密钥协商过程创建临时密钥。该命令接口输入和输出具体参数要求见表 71 和表 72。

表 71 TCM2_EC_Ephemeral()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_EC_Ephemeral
TCMI_ECC_CURVE	curveID	用于计算短暂密钥的曲线

表 72　TCM2_EC_Ephemeral()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	Q	短暂公钥 Q:=[r]G
UINT16	counter	commitCount 的低 16 比特

7.12.2　TCM2_Commit

TCM2_Commit()执行 ECC 匿名签名操作的第一部分。TCM 将对提供的点执行点乘法,并返回中间签名值。signHandle 参数应引用 ECC 密钥,签名方案应是匿名的(TCM2_RC_scheme)。

注:此命令不能与 sign+decrypt 密钥一起使用,因为该类型的密钥应具有 TCM2_ALG_NULL 的方案。

对于该命令,p1、s2 和 y2 是可选参数。如果 s2 是空缓冲区,在 y2 不是空缓冲区的情况下,TCM 应返回 TCM2_RC_SIZE。

该命令接口输入和输出具体参数要求见表 73 和表 74。

表 73　TCM2_Commit()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Commit
TCMI_DH_OBJECT	@signHandle	用于签名操作的密钥句柄 授权索引:1 授权角色:USER
TCM2B_ECC_POINT	p1	在曲线上被 signHandle 使用的一个点(M)
TCM2B_SENSITIVE_DATA	s2	用于导出基点 x 坐标的八进制数组
TCM2B_ECC_PARAMETER	y2	与 s2 关联点的 y 坐标

表 74　TCM2_Commit()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_ECC_POINT	K	ECC 上的点 K:=[d_s](x2,y2)
TCM2B_ECC_POINT	L	ECC 上的点 L:=[r](x2,y2)
TCM2B_ECC_POINT	E	ECC 上的点 E:=[r]P1
UINT16	counter	计数器的最低有效 16 比特

7.13 签名及签名验证命令

7.13.1 TCM2_VerifySignature

该命令使用一个已加载的密钥来验证一个消息的签名，该消息与其摘要一起传给 TCM。如果签名验证成功，则 TCM 返回 TCMT_TK_VERIFIED，否则 TCM 返回 TCM2_RC_SIGNATURE。

SM2 签名及签名验证功能应符合 GB/T 32918.2，使用规范应符合 GB/T 35276。

该命令接口输入和输出具体参数要求见表 75 和表 76。

表 75 TCM2_VerifySignature()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密会话，应为 TCM2_ST_SESSIONS，否则应为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_VerifySignature
TCMI_DH_OBJECT	keyHandle	用来验证签名的密钥的公开部分 授权索引：None
TCM2B_DIGEST	digest	被签名消息的摘要
TCMT_SIGNATURE	signature	将要被验证的签名

表 76 TCM2_VerifySignature()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMT_TK_VERIFIED	validation	验证结果

7.13.2 TCM2_Sign

该命令使 TCM 对一个外部提供的杂凑值用特定的非对称签名密钥进行签名。签名及签名验证应符合 GB/T 32918.2，使用规范应符合 GB/T 35276。

a) 如果签名密钥是受限的，那么有效的验证被应提供来表明待签名的数据杂凑值时由 TCM 产生且被计算杂凑的数据不是由 TCM2_GENERATED_VALUE 开始的；

b) 如果签名密钥的签名方案不是 TCM2_ALG_NULL，那么 inScheme 参数指定的签名方案或者与密钥签名方案相同或者是 TCM2_ALG_NULL；

c) 如果密钥使用匿名签名方案，那么 inScheme 应指定同样的签名算法，且应包含一个计数器，将在签名过程中使用；

d) 如果验证参数 validation 不是 Empty buffer，那么即使签名密钥不是受限的，验证信息也将被检查。

该命令接口输入和输出具体参数要求见表 77 和表 78。

表 77　TCM2_Sign()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Sign
TCMI_DH_OBJECT	@keyHandle	签名密钥句柄 授权索引:1 授权角色:USER
TCM2B_DIGEST	digest	要被签名的摘要值
TCMT_SIG_SCHEME+	inScheme	如果密钥的签名方案是 TCM2_ALG_NULL,则此参数指定签名方案
TCMT_TK_HASHCHECK	validation	证明杂凑值是由 TCM 创建的

表 78　TCM2_Sign()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMT_SIGNATURE	signature	签名结果

7.14　度量命令

7.14.1　TCM2_PCR_Extend

该命令用于更新指定的 PCR 值。digests 参数中包含一个或多个通过算法 ID 绑定的杂凑值。对每个杂凑值,扩展到 pcrHandle 指向的 PCR 中。

对每一个入口,TCM 将会检测 pcrNum 是否实现了指定的算法。如果实现了,TCM 将会执行下面的操作:

$$PCR.digestnew[pcrNum][alg] := H_{alg}(PCR.digestold[pcrNum][alg] \| data[alg].buffer))$$

如果指定的 PCR 没有提供摘要值,那么该 PCR 的值将不会改变。

该命令接口输入和输出具体参数要求见表 79 和表 80。

表 79　TCM2_PCR_Extend()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PCR_Extend {NV}

表 79 TCM2_PCR_Extend ( ) 接口输入参数(续)

类型	名称	描述
TCMI_DH_PCR+	@pcrHandle	PCR句柄 授权句柄:1 授权角色:USER
TCML_DIGEST_VALUES	digests	待扩展的摘要值列表

表 80 TCM2_PCR_Extend()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.14.2 TCM2_PCR_Read

该命令返回 pcrSelect 中指定的所有 PCR 的值。

TCM 将按序处理 pcrSelectionIn 中的 TCMS_PCR_SELECTION 列表。TCM 将会按照 PCR 升序按位处理 pcrSelect 数组。如果位被设置,且当前指定的 PCR 存在,TCM 将会增加 PCR 摘要值到 pcrValue 中并且返回该值。

TCM 将依序处理每一位表示的 PCR,当 pcrValues 指定的位处理完全或者 pcrValues 中没有足够的空间填充输出数据。

该命令接口输入和输出具体参数要求见表 81 和表 82。

表 81 TCM2_PCR_Read()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PCR_Read
TCML_PCR_SELECTION	pcrSelectionIn	被选择执行读操作的PCR

表 82 TCM2_PCR_Read()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
UINT32	pcrUpdateCounter	PCR更新计数器中的当前值
TCML_PCR_SELECTION	pcrSelectionOut	返回列表中的PCR
TCML_DIGEST	pcrValues	pcrSelect中指定PCR的摘要值

7.14.3 TCM2_PCR_Reset

如果一个 PCR 的属性允许该 PCR 使用恰当的授权值进行重置操作，那么该命令可被用于设置 PCR 的值为零。PCR 的属性可能会限制执行重置操作的位置。

如果 pcrHandle 参数指向的 PCR 不能执行重置，那么 TCM 将会返回无法重置。

该命令接口输入和输出具体参数要求见表 83 和表 84。

表 83 TCM2_PCR_Reset()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PCR_Reset {NV}
TCMI_DH_PCR	@pcrHandle	被重置的 PCR 句柄 授权索引：1 授权角色：USER

表 84 TCM2_PCR_Reset()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15 增强授权命令

7.15.1 TCM2_PolicySigned

该命令是在增强策略中包含一个签名授权。该命令是通过在 policyDigest 中包含签名密钥的名称的方式来实现增强策略与签名密钥的绑定。

如果 policySession 是一个试用会话(trial session)，TCM 将不会检测签名值，如果收到正确的签名授权值，TCM 将会更新 policySession→policyDigest 的值。如果 policySession 不是试用会话，TCM 将会验证授权，如果是一个有效签名，TCM 只会执行更新操作。

授权对象将会对如下授权限定词的摘要值进行签名：nonceTCM，expiration，cpHashA，以及 policyRef。摘要计算公式如下：

$$aHash := H_{authAlg}(nonceTCM \| expiration \| cpHashA \| policyRef)$$

该命令接口输入和输出具体参数要求见表 85 和表 86。

表 85 TCM2_PolicySigned()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密或者解密，则是 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicySigned
TCMI_DH_OBJECT	authObject	验证签名的公钥句柄 授权索引：None
TCMI_SH_POLICY	policySession	扩展的策略会话句柄 授权索引：None
TCM2B_NONCE	nonceTCM	会话的策略 nonce 值 如果该 nonce 值不包含在授权资格中，该字段为空
TCM2B_DIGEST	cpHashA	该授权限制的命令参数摘要 该值不是命令所需的 cpHash，而是策略会话被应用的命令的 cpHash 值，如果没有限制，该参数可为空
TCM2B_NONCE	policyRef	指向于授权关联的策略，可为空 大小不能大于 TCM 支持的 nonce 的长度大小
INT32	expiration	授权将过期的时间，从 nonceTCM 创建开始以秒为单位进行计时 如果 expiration 是 0，没有票据返回
TCMT_SIGNATURE	auth	被签名的授权(非可选)

表 86 TCM2_PolicySigned()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_TIMEOUT	timeout	执行特定时间值，当票据过期时用于指向 TCM 如果策略票据是空票据，该项为空。
TCMT_TK_AUTH	policyTicket	命令操作成功且命令中的 expiration 值是非零，该票据使用 TCMT_ST_AUTH_SIGNED 结构的标签

7.15.2 TCM2_PolicySecret

该命令包含基于秘密授权的增强策略。

调用方使用与 authhandle 关联的 authvalue,通过授权会话来证明对秘密值的了解。授权会话使用与 authHandle 关联的 authValue。不管是 password 会话,HMAC 会话或者是包含 TCM2_PolicyhAuthValue()或者 TCM2_PolicyPassword()命令的增强策略会话应满足该需求。

如果使用了策略会话,但 authHandle 的 authValue 不是所需的,TCM 将会返回 TCM2_RC_MODE。

该命令接口输入和输出具体参数要求见表 87 和表 88。

表 87 TCM2_PolicySecret()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicySecret
TCMI_DH_ENTITY+	@authHandle	提供授权实体的句柄 授权句柄:1 授权角色:USER
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引:None
TCM2B_NONCE	nonceTCM	会话的策略 nonce 如果 nonce 没有包含在授权资格中,该域为空
TCM2B_DIGEST	cpHashA	该授权限制的命令参数摘要 该值不是命令所需的 cpHash,而是策略会话被应用的命令的 cpHash 值,如果没有限制,该参数可为空
TCM2B_NONCE	policyRef	指向授权关联的策略,可为空 大小不能大于 TCM 支持的 nonce 的大小
INT32	expiration	授权将过期的时间,从 nonceTCM 创建开始以秒(s)为单位进行计时 如果 expiration 是零,没有票据返回

表 88 TCM2_PolicySecret()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_TIMEOUT	timeout	执行特定时间值,当票据过期时用于指向 TCM 如果策略票据是空票据,该项为空
TCMT_TK_AUTH	policyTicket	命令操作成功且命令中的 expiration 值是非零,该票据使用 TCMT_ST_AUTH_SIGNED 结构的标签

7.15.3 TCM2_PolicyTicket

该命令与 TCM2_PolicySigned()类似,使用了票据替换了签名授权。票据代表一个有有效期的经过验证的授权。

如果检测成功,TCM 使用 timeout,cpHashA,policyRef,以及 keyName 创建一个票据信息,与输入的票据信息进行比较,如果匹配,TCM 将使用 authName 以及使用 PolicyUpdate()更新的 policySession 的上下文创建一个 TCM2B_NAME(objectName)。

PolicyUpdate(commandCode, authName, policyRef)

如果票据的结构 tag 是 TCM2_ST_AUTH_SECRET,commandCode 将是 TCM2_CC_PolicySecret。如果票据的结构 tag 是 TCM2_ST_AUTH_SIGNED,commandCode 将是 TCM2_CC_PolicySigned。

如果 cpHashA 参数不为空,这个值将会复制到 cpHash 中。

该命令接口输入和输出具体参数要求见表 89 和表 90。

表 89 TCM2_PolicyTicket()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是是解密会话,则是 TCM2_ST_SESSIONS,否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyTicket
TCMI_SH_POLICY	policySession	扩展的策略会话句柄 授权索引:None
TCM2B_TIMEOUT	timeout	授权将要过期的时间 内容由 TCM 指定,可以是票据产生时返回的值
TCM2B_DIGEST	cpHashA	该授权限制的命令参数摘要 该值不是命令所需的 cpHash,而是策略会话被应用的命令的 cpHash 值,如果没有限制,该参数可为空
TCM2B_NONCE	policyRef	指向于授权关联的策略—可为空 大小不能大于 TCM 支持的 nonce 长度大小
TCM2B_NAME	authName	提供授权对象的名称
TCMT_TK_AUTH	ticket	TCM2_PolicySigned () 或 TCM2_PolicySecret () 返回 TCM 授权票据

表 90 TCM2_PolicyTicket()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.4 TCM2_PolicyOR

该命令允许授权中的选项，无需 TCM 评估所有选项。如果一个策略可以由不同的条件集满足，TCM 只评估满足该策略的一个集合。该命令将指示已满足所需条件集之一。

PolicySession→policyDigest 与被提供的值进行比较。如果当前 policySession→policyDigest 与列表中的任意值都不匹配，TCM 将返回 TCM2_RC_VALUE。否则，会串联所有的摘要值进行摘要运算，用该计算的摘要值替换 policySession→policyDigest 的值并返回 TCM2_RC_SUCCESS。

如果 policySession 是一个试用会话，TCM 将会假设 policySession→policyDigest 与列表中任意一个值匹配，并计算出新的 policyDigest。

该命令接口输入和输出具体参数要求见表 91 和表 92。

表 91 TCM2_PolicyOR( )接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyOR
TCMI_SH_POLICY	policySession	待扩展策略会话句柄 授权索引：None
TCML_DIGEST	pHashList	匹配检测的杂凑列表

表 92 TCM2_PolicyOR( )接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.5 TCM2_PolicyPCR

该命令被用于触发基于 PCR 的条件策略。

当 PCR 处于一种状态时，允许一组授权的发生。当 PCR 处于不同状态时，允许不同集合的授权。如果命令被用于试用策略会话，policySession→policyDigest 将使用命令中使用的值进行更新，而不是 TCM PCR 的摘要值。

TCM 将会改变 pcrs 参数的值，与未实现的 PCR 相对应的位将会被清空。如果 policySession 不是试用策略会话，TCM 将会根据已改变的 pcrs 的值去选择 PCR 对应的值进行杂凑计算。使用策略会话指定的 SM3 杂凑算法对选择的 PCR 进行摘要计算(digestTCM)。如果 pcrDigest 的长度不为零，应比较 digestTCM 与 pcrDigest 的值，如果不匹配，TCM 将返回 TCM2_RC_VALUE，同时不改变 policySession→policyDigest 的值。如果匹配，或者 pcrDigest 的长度为零，policySession→policyDigest 将被扩展：

$$policyDigest_{new} := H_{policyAlg}(policyDigest_{old} \| TCM2\_CC\_PolicyPCR \| pcrs \| digestTCM)$$

该命令接口输入和输出具体参数要求见表 93 和表 94。

表 93 TCM2_PolicyPCR()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果解密，则是 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyPCR
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引：None
TCM2B_DIGEST	pcrDigest	选择的PCR使用会话中的杂凑算法得到期望的杂凑值可能为零长度
TCML_PCR_SELECTION	pcrs	包含在检测摘要中的PCR

表 94 TCM2_PolicyPCR()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.6 TCM2_PolicyCommandCode

该命令表明授权被限制于特定命令。

如果 policySession→commandCode 有默认值，那么它将会被设置为 code 值，如果 policySession→commandCode 没有默认值，如果两者值不同时，TCM 将会返回 TCM2_RC_VALUE。

如果 code 没有实现，TCM 将会返回 TCM2_RC_POLICY_CC。如果 TCM 没有返回错误，policySession→policyDigest 将会采用如下值更新：

$policyDigest_{new} := H_{policyAlg}(policyDigest_{old} \| TCM2\_CC\_PolicyCommandCode \| code)$

当策略会话被用于授权命令，如果 commandCode 与 policySession→commandCode 不匹配，该命令将会失败。

该命令接口输入和输出具体参数要求见表 95 和表 96。

表 95 TCM2_PolicyCommandCode()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyCommandCode
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引：None
TCM2_CC	code	允许的命令代码

表 96 TCM2_PolicyCommandCode ()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.7 TCM2_PolicyPhysicalPresence

该命令表明在执行授权时应声明物理存在。

如果该命令成功，policySession→isPPRequired 将会被设置，这表明该策略被用于授权时应被检测。policySession→policyDigest 使用如下值进行扩展：

$policyDigest_{new}:=H_{policyAlg}(policyDigest_{old} \| TCM2\_CC\_PolicyPhysicalPresence)$

该命令接口输入和输出具体参数要求见表 97 和表 98。

表 97 TCM2_PolicyPhysicalPresence()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyPhysicalPresence
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引：None

表 98 TCM2_PolicyPhysicalPresence()接口输出参数

类型	名称	描述
TCM2_ST	Tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.8 TCM2_PolicyCpHash

该命令被用于允许策略与特定的命令及命令参数绑定。

当该命令的 cpHashA 参数没有被包含在 policySession→PolicyDigest 中，TCM2_PolicySigned()，TCM2_PolicySecret()及 TCM2_PolicyTicket 被设计于可允许授权实体执行任意命令。TCM2_PolicyCommandCode()允许策略与特定的命令代码绑定，所以只有特定的实体可授权特定命令代码。

如果 policySession→cpHash 已经设置且与 cpHashA 不一样，TCM 将会返回 TCM2_RC_VALUE。如果 cpHashA 与 policySession→policyDigest 大小不一样，TCM 将会返回 TCM2_RC_SIZE。

如果 cpHahsA 检测成功，policySession→cpHash 设置为 cpHashA 值，且 policySession→policyDigest 采用如下值进行更新：

$policyDigest_{new}:=H_{policyAlg}(policyDigest_{old} \| TCM2\_CC\_PolicyCpHash \| cpHashA)$

该命令接口输入和输出具体参数要求见表 99 和表 100。

表 99　TCM2_PloicyCpHash()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是解密,则是 TCM2_ST_SESSIONS,否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyCpHash
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引:None
TCM2B_DIGEST	cpHashA	加到策略的　cpHash

表 100　TCM2_PloicyCpHash()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.9　TCM2_PolicyAuthValue

该命令允许将策略绑定到授权对象的授权值。

如果该命令成功完成,policySession→isAuthValueNeeded 将会被设置,用于表明当计算会话的授权 HMAC 值,authValue 值将被包含在 hmacKey 中。另外,policySession→isPasswordNeeded 值将被清除。

如果操作成功,policySession→policyDigest 会使用如下值进行更新:

$policyDigest_{new} := H_{policyAlg}(policyDigest_{old} \| TCM2\_CC\_PolicyAuthValue)$

该命令接口输入和输出具体参数要求见表 101 和表 102。

表 101　TCM2_PolicyAuthValue()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyAuthValue
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引:None

表 102 TCM2_PolicyAuthValue()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.10 TCM2_PolicyPassword

该命令允许策略与已授权对象的授权值绑定。

当命令成功完成，policySession→isPasswordNeeded 被设置，用于表明当授权会话被使用时，已授权对象的 authvalue 将被检测。调用者将在授权的 hmac 参数中提供 authValue。当授权方式是口令(password)方式时，将比较 hmac 与 authValue 的值。

该命令接口输入和输出具体参数要求见表 103 和表 104。

表 103 TCM2_PloicyPassword()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PolicyPassword
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引：None

表 104 TCM2_PloicyPassword()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.15.11 TCM2_PolicyGetDigest

该命令返回当前会话的 policyDigest 的值。该命令允许 TCM 被用于提前为对象计算 authPolicy 值。

该命令接口输入和输出具体参数要求见表 105 和表 106。

表 105 TCM2_PloicyGetDigest()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	如果是加密，则是 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度

表 105 TCM2_PloicyGetDigest（）接口输入参数（续）

类型	名称	描述
TCM2_CC	commandCode	TCM2_CC_PolicyGetDigest
TCMI_SH_POLICY	policySession	被扩展策略会话的句柄 授权索引：None

表 106 TCM2_PloicyGetDigest()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_DIGEST	policyDigest	policySession→policyDigest 的当前值

7.16 分层命令

7.16.1 TCM2_CreatePrimary

该命令被用来在一个主种子下创建主要对象，或者 TCM2_RH_NULL 的临时对象，该命令使用输入参数 TCM2B_PUBLIC 作为创建对象的模板。该命令将创建并加载一个主要对象，对象的敏感数据不会被返回。

该命令创建 TCM2_Create()允许的任何类型的对象和属性组合，对模板和参数的约束与限制与 TCM2_CREATE 相同，只是主存储密钥和临时存储密钥不受约束以使用其父密钥算法。

该命令要求相应的授权信息，与 PPS 关联的主对象要求提供 platformAuth 或 platformPolicy 授权信息，与 SPS 关联的主要对象要求调用者提供 ownerAuth 或 ownerPolicy 授权信息，与 EPS 关联的主要对象要求调用者提供 endorsement 或 endorsementPolicy 授权信息。

该命令接口输入和输出具体参数要求见表 107 和表 108。

表 107 TCM2_CreatePrimary()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_CreatePrimary
TCMI_RH_HIERARCHY+	@primaryHandle	TCM2_RH_ENDORSEMENT， TCM2_RH_OWNER， TCM2_RH_PLATFORM+{PP}或者 TCM2_RH_NULL 授权索引：1 授权角色：USER
TCM2B_SENSITIVE_CREATE	inSensitive	隐私数据部分

表 107 TCM2_CreatePrimary（）接口输入参数（续）

类型	名称	描述
TCM2B_PUBLIC	inPublic	公开区域模板
TCM2B_DATA	outsideInfo	该参数将被包含到输出参数 creationData 中，提供被创建对象与创建者之间永久可验证的联系
TCML_PCR_SELECTION	creationPCR	指定被使用的 PCR 寄存器

表 108 TCM2_CreatePrimary()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2_HANDLE	objectHandle	创建的主对象句柄
TCM2B_PUBLIC	outPublic	创建对象的公开区域
TCM2B_CREATION_DATA	creationData	返回的创建数据
TCM2B_DIGEST	creationHash	创建数据的摘要值
TCMT_TK_CREATION	creationTicket	验证创建数据确实是由 TCM 产生的票据凭证
TCM2B_NAME	name	被创建对象的名字

7.16.2 TCM2_HierarchyControl

该命令开启或禁止对一个 hierarchy 等级的使用，当合适的授权信息被提供时，phEnable，shEnable，ehEnable 将允许被改变。当该命令用来禁用一个 hierarchy 等级时，TCM 将禁用所有与该等级关联的永久实体，且将清除所有与该等级关联的临时对象。

该命令接口输入和输出具体参数要求见表 109 和表 110。

表 109 TCM2_HierarchyControl()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_HierarchyControl{NV E}
TCMI_RH_HIERARCHY	@authHandle	TCM2_RH_ENDORSEMENT， TCM2_RH_OWNER 或者 TCM2_RH_PLATFORM+{PP} 授权索引：1 授权身份：USER

表 109 TCM2_HierarchyControl( ) 接口输入参数 (续)

类型	名称	描述
TCMI_RH_ENABLES	Enable	将被改变的 hierarchy 等级 TCM2_RH_ENDORSEMENT, TCM2_RH_OWNER 或 TCM2_RH_PLATFORM
TCMI_YES_NO	state	指定此次调用时启用还是禁用命令

表 110 TCM2_HierarchyControl( )接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.16.3 TCM2_SetPrimaryPolicy

该命令允许设置 platform,storage,endorsement 等级的授权策略。该命令要求一个授权会话,该会话应使用当前的授权值,或者满足当前的授权策略。参数 authHandle 指定了将要改变授权策略的 hierarchy 等级。如果 authHandle 指定的 hierarchy 没有被启用,那么关联的授权信息将不被使用。

该命令接口输入和输出具体参数要求见表 111 和表 112。

表 111 TCM2_SetPrimaryPolicy( )接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_SetPrimaryPolicy{NV}
TCMI_RH_HIERARCHY_POLICY	@authHandle	TCM2_RH_ENDORSEMENT,TCM2_RH_OWNER 或者 TCM2_RH_PLATFORM+{PP} 授权索引:1 授权身份:USER
TCM2B_DIGEST	authPolicy	授权策略的摘要值,如果 hashAlg 参数为空,则此参数应为 Empty Buffer
TCMI_ALG_HASH+	hashAlg	计算摘要使用的 hash 算法,如果参数 authPolicy 为空,则此参数应为 TCM2_ALG_NULL

表 112　TCM2_SetPrimaryPolicy()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据长度
TCM2_RC	responseCode	返回码

7.16.4　TCM2_Clear

该命令移除与指定所有者关联的所有 TCM 上下文，要求 platformAuth 和 lockoutAuth 授权信息。

如果 TCM2_ClearControl 已经禁用了该命令，则 TCM 返回 TCM2_RC_DISABLED 。

如果使用 lockoutAuth 授权信息，则响应 HMAC 将使用新的 lockoutAuth 授权值(Empty Buffer)计算。

该命令接口输入和输出具体参数要求见表 113 和表 114。

表 113　TCM2_Clear()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_Clear{NV E}
TCMI_RH_CLEAR	@authHandle	TCM2_RH_LOCKOUT 或者 TCM2_RH_PLATFORM+{PP} 授权索引：1 授权身份：USER

表 114　TCM2_Clear()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.16.5　TCM2_ClearControl

该命令启用或禁用 TCM2_Clear 命令的执行。

参数 disable 指定此次调用是启用还是禁用 TCM2_Clear 命令。

使用 LockoutAuth 授权信息用于设置 disableClear 但不能清除。

使用 PlatformAuth 授权信息可禁用或启用 TCM2_Clear 命令。

该命令接口输入和输出具体参数要求见表 115 和表 116。

表 115 TCM2_ClearControl()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ClearControl
TCMI_RH_CLEAR	@auth	TCM2_RH_LOCKOUT or TCM2_RH_PLATFORM+{PP} 授权索引:1 授权身份:USER
TCMI_YES_NO	disable	YES:disableOwnerClear标志设置成SET NO:disableOwnerClear标志设置成CLEAR

表 116 TCM2_ClearControl()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据长度
TCM2_RC	responseCode	返回码

7.16.6 TCM2_HierarchyChangeAuth

该命令将改变一个 hierarchy 等级或 lockout 的授权秘密值,使用当前的授权值作为命令授权。新的授权值长度不能大于用于计算上下文完整性的 SM3 杂凑算法产生的摘要值长度。

该命令接口输入和输出具体参数要求见表 117 和表 118。

表 117 TCM2_HierachyChangeAuth()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_HierarchyChangeAuth{NV}
TCMI_RH_HIERARCHY_AUTH	@authHandle	TCM2_RH_LOCKOUT,TCM2_RH_ENDORSEMENT,TCM2_RH_OWNER 或M_RH_PLATFORM+{PP} 授权索引:1 授权身份:USER
TCM2B_AUTH	newAuth	新的授权值

表 118 TCM2_HierachyChangeAuth()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据长度
TCM2_RC	responseCode	返回码

7.17 字典攻击命令

7.17.1 TCM2_DictionaryAttackLockReset

当一定数量的连续授权失败,TCM 将会被锁定,该命令用于退出该锁定模式。如果该命令被正确授权,锁定计数器(lockout counter)设置零。

在一个 lockoutRecovery 间隔,仅允许一次授权失败。

该命令接口输入和输出具体参数要求见表 119 和表 120。

表 119 TCM2_DictionaryAttackLockReset()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_DictionaryAttackLockReset {NV}
TCMI_RH_LOCKOUT	@lockHandle	TCM2_RH_LOCKOUT 授权索引:1 授权角色:USER

表 120 TCM2_DictionaryAttackLockReset()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.17.2 TCM2_DictionaryAttackParameters

该命令用于改变锁定参数。

a) 使用 lockoutAuth。

b) 超时参数(newRecoveryTime 和 lockoutRecovery)表明测量的值需考虑时间而非时钟。
该命令设置授权失败计数器(failedTries)为零。

c) 在 lockoutRecovery 期间,该命令只允许一次授权失败。

该命令接口输入和输出具体参数要求见表 121 和表 122。

表 121 TCM2_DictionaryAttackParameters()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_DictionaryAttackParameters {NV}
TCMI_RH_LOCKOUT	@lockHandle	TCM2_RH_LOCKOUT 授权索引：1 授权角色：USER
UINT32	newMaxTries	被锁定之前支持的最大失败次数
UINT32	newRecoveryTime	失败计数器自动减少前的秒数
UINT32	lockoutRecovery	允许使用 lockoutAuth 之前，lockoutAuth 失败之后的时间秒数 零值表示应重启

表 122 TCM2_DictionaryAttackParameters()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.18 管理功能命令

TCM2_PP_Commands 命令被用于定义作为对 platformAuth/platformPolicy 的补充，表明哪些命令依赖物理存在。

该命令 auth 是 TCM2_RH_PLATFORM，且声明物理存在。

该命令执行成功，出现在 setList 中的命令将被添加到与句柄关联的授权(TCM2_RH_PLATFORM)声明的物理存在的命令列表中。clearList 列表中的命令不再声明物理存在。

如果一个命令没有出现在任意一个列表中，其状态不会被改变。如果一个命令出现在两个列表中，它将不再声明物理存在。

声明物理现场存在成功之后，TCM2_PP_Command()命令有效。

该命令接口输入和输出具体参数要求见表 123 和表 124。

表 123 TCM2_PP_Commands()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_PP_Commands {NV}

表 123 TCM2_PP_Commands ( ) 接口输入参数 (续)

类型	名称	描述
TCMI_RH_PLATFORM	@auth	TCM2_RH_PLATFORM+PP 授权索引：1 授权角色：USER+Physical Presence
TCML_CC	setList	增加物理存在声明的命令列表
TCML_CC	clearList	清除物理存在声明的命令列表

表 124 TCM2_PP_Commands()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.19 上下文管理命令

7.19.1 TCM2_ContextSave

该命令用于保存会话上下文或序列对象上下文。该命令是非授权的会话，tag 应是 TCM2_ST_NO_SESSIONS。TCM 将会加密及完整性保护上下文。

该命令接口输入和输出具体参数要求见表 125 和表 126。

表 125 TCM2_ContextSave()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ContextSave
TCMI_DH_CONTEXT	saveHandle	保存资源的句柄 授权索引：None

表 126 TCM2_ContextSave()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMS_CONTEXT	context	上下文数据块

7.19.2 TCM2_ContextLoad

该命令用于重新加载使用 TCM2_ContextSave()命令保存的上下文。

该命令是非授权的会话,tag 应是 TCM2_ST_NO_SESSIONS。

当与上下文关联的分层管理是作废的,TCM 将会返回 TCM2_RC_HIERARCHY。

该命令接口输入和输出具体参数要求见表 127 和表 128。

表 127 TCM2_ContextLoad()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_ContextLoad
TCMS_CONTEXT	context	上下文数据块

表 128 TCM2_ContextLoad()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMI_DH_CONTEXT	loadedHandle	成功加载后指向资源的句柄

7.19.3 TCM2_FlushContext

该命令把与导入对象或者会话关联的上下文都从 TCM 内存中清除。

该命令不能用于从 TCM 中清除持久存储对象。

该命令是非授权的会话,tag 应是 TCM2_ST_NO_SESSIONS。

如果句柄为临时对象,且句柄没有与加载的对象关联,TCM 将返回 TCM2_RC_HANDLE。

如果句柄指向的是授权会话,且该句柄没有与加载或者激活的会话关联,TCM 将返回 TCM2_RC_HANDLE。

该命令接口输入和输出具体参数要求见表 129 和表 130。

表 129 TCM2_FlushContext()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_FlushContext
TCMI_DH_CONTEXT	flushHandle	待释放的句柄

表 130 TCM2_FlushContext()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.19.4 TCM2_EvictControl

该命令允许临时对象变为持久对象，或者一个持久对象被驱赶出 TCM。

如果 objectHandle 是临时对象，该调用使对象持久存储且分配一个持久句柄。如果 objectHandle 是一个持久对象，该调用将驱赶出该持久对象。

该命令接口输入和输出具体参数要求见表 131 和表 132。

表 131 TCM2_EvictControl()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_EvictControl {NV}
TCMI_RH_PROVISION	@auth	TCM2_RH_OWNER or TCM2_RH_PLATFORM+{PP} 授权句柄：1 授权角色：USER
TCMI_DH_OBJECT	objectHandle	被加载对象句柄 授权索引：None
TCMI_DH_PERSISTENT	persistentHandle	如果 objectHandle 是一个短暂对象句柄，那么这是一个为该对象指定的持久句柄。 如果 objectHandle 是持久对象句柄，那么该值与 objectHandle 值一致

表 132 TCM2_EvictControl()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.20 属性命令

7.20.1 TCM2_GetCapability

该命令根据 TCM 及其当前状态返回不同的信息。

Capability 参数定义了返回数据的类型。Property 参数选择被选择返回类的第一个值。如果没有与 property 值关联的特性值,但如果存在下一个高值则返回该值。

propertyCount 参数表明在指定的组中对应的个数。TCM 将会返回对应属性的个数(propertyCount) 或者返回对应类型的所有的特性。

当所有特性已被返回,moreData 参数设置为 NO,否则设置为 YES。

该命令接口输入和输出具体参数要求见表 133 和表 134。

表 133 TCM2_GetCapability()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_GetCapability
TCM2_CAP	capability	分组选择,确定了返回值的格式
UINT32	property	信息的更深定义
UINT32	propertyCount	指明类型返回的属性数量

表 134 TCM2_GetCapability()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCMI_YES_NO	moreData	表示是否还有更多的该类型的属性值
TCMS_CAPABILITY_DATA	capabilityData	能力数据

7.20.2 TCM2_TestParms

该命令用于检测特定配合的算法参数是否被支持。TCM 将会对提供的 TCMT_PUBLIC_PARAMS 参数进行解析。如果参数解析成功,TCM 将会返回 TCM2_RC_SUCCESS,表明对 TCM 来说该参数是有效的。如果参数无效,TCM 将会返回恰当的解析错误。

该命令接口输入和输出具体参数要求见表 135 和表 136。

表 135 TCM2_TestParms()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_TestParms
TCMT_PUBLIC_PARMS	parameters	待验证的算法参数

表 136 TCM2_TestParms()接口输出参数

类型	姓名	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21 NV操作命令

7.21.1 TCM2_NV_DefineSpace

该命令为 NV 索引定义属性，TCM 预留用于保留与 NV 索引关联数据的空间。如果对一个已经存在的索引执行定义操作，TCM 将返回 TCM2_RC_DEFINED。

a) 如果 publicInfo 中有多余一个 TCMA_NV_COUNTER，TCMA_NV_BITS，或者 TCMA_NV_EXTEND 被设置，TCM 将会返回 TCM2_RC_ATTRIBUTES。

b) 如果 TCMA_NV_COUNTER 或 TCMA_NV_BITS 被设置，publicInfo→dataSize 将被设置为 8，否则 TCM 将返回 TCM2_RC_SIZE。

c) 如果 TCMA_NV_EXTEND 被设置，publicInfo→dataSize 需匹配 publicInfo.nameAlg 的摘要大小，否则 TCM 返回 TCM2_RC_SIZE。

d) 如果 NV 索引是普通索引，publicInfo→dataSize 大于 TCM 支持的最大值，TCM 将返回 TCM2_RC_SIZE。

auth 的值被保存在创建的结构中。auth 的长度不能大于 NV 索引的 nameAlg(TCM2_RC_SIZE)产生的摘要值。

该命令接口输入和输出具体参数要求见表 137 和表 138。

表 137 TCM2_NV_DefineSpace()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_DefineSpace {NV}
TCMI_RH_PROVISION	@authHandle	TCM2_RH_OWNER or TCM2_RH_PLATFORM+{PP} 授权索引:1 授权角色:USER
TCM2B_AUTH	auth	授权值
TCM2B_NV_PUBLIC	publicInfo	NV区的公共参数

表 138 TCM2_NV_DefineSpace()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.2 TCM2_NV_UndefineSpace

该命令从 TCM 中移除索引。

如果 nvIndex 没有定义,TCM 将会返回 TCM2_RC_HANDLE。如果 nvIndex 指向的索引的 TCMA_NV_PLATFORMCREATE 属性被设置,如果没有提供 platformAuth,TCM 将返回 TCM2_RC_NV_AUTHORITY。

该命令接口输入和输出具体参数要求见表 139 和表 140。

表 139 TCM2_NV_UndefineSpace()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_UndefineSpace {NV}
TCMI_RH_PROVISION	@authHandle	TCM2_RH_OWNER or TCM2_RH_PLATFORM+{PP} 授权索引:1 授权角色:USER
TCMI_RH_NV_INDEX	nvIndex	从NV空间移除的NV索引 授权索引:None

表 140 TCM2_NV_UndefineSpace()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.3 TCM2_NV_ReadPublic

该命令被用于读取给定 NV 索引的公共区及 NV 索引的名字。索引的公共区不是私有敏感区,非受权的方式读取这些数据。

该命令接口输入和输出具体参数要求见表 141 和表 142。

表 141 TCM2_NV_ReadPublic()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_NO_ESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_ReadPublic
TCMI_RH_NV_INDEX	nvIndex	NV索引 授权索引：None

表 142 TCM2_NV_ReadPublic()接口输出参数

类型	名称	描述
TCM2_ST	tag	TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_NV_PUBLIC	nvPublic	指定NV索引的NV公共区
TCM2B_NAME	nvName	nvIndex的名称

7.21.4 TCM2_NV_Write

该命令用于往使用 TCM2_NV_DefineSpace()命令定义的 NV 空间写入值。该命令应使用正确的授权，如：TCMA_NV_PPWRITE、TCMA_NV_OWNERWRITE、TCMA_NV_AUTHWRITE。

如果 NV 索引的 TCMA_NV_WRITELOCKED 属性设置，TCM 将会返回 TCM2_RC_NV_LOCKED。

该命令接口输入和输出具体参数要求见表 143 和表 144。

表 143 TCM2_NV_Write()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	Tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_Write
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	待写NV区的索引 授权索引：None
TCM2B_MAX_NV_BUFFER	data	待写入NV区的数据
UINT16	offset	NV区的偏移量

表 144 TCM2_NV_Write()接口输出参数

类型	名称	描述
TCM2_ST	Tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.5 TCM2_NV_Increment

该命令被用于为 NV 索引中增加值,该 NV 索引应满足 TCMA_NV_COUNTER 属性被设置。NV 索引增加 1。

如果指定的 NV 索引的 TCMA_NV_COUNTER 没有设置,TCM 将返回 TCM2_RC_ATTRIBUTES。

如果 TCMA_NV_WRITELOCKED 被设置,TCM 将返回 TCM2_RC_NV_LOCKED。

该命令接口输入和输出具体参数要求见表 145 和表 146。

表 145 TCM2_NV_Increment()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_Increment
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引:1 授权角色:USER
TCMI_RH_NV_INDEX	nvIndex	待增加的NV索引 授权索引:None

表 146 TCM2_NV_Increment()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.6 TCM2_NV_Extend

该命令往使用 TCM2_NV_DefineSpace 命令定义的 NV 区域扩展值。

如果 TCMA_NV_EXTEND 没有设置,TCM 将会返回 TCM2_RC_ATTRIBUTES。

该命令使用正确的授权，命令成功完成后，NV 索引的 TCMA_NV_WRITEN 将被设置。

如果 NV 索引的 TCMA_NV_WRITELOCKED 属性被设置，TCM 将会返回 TCM2_RC_NV_LOCKED。

data.buffer 参数可大于 NV 索引定义的 NV 空间大小。

该命令接口输入和输出具体参数要求见表 147 和表 148。

表 147　TCM2_NV_Extend()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_Extend {NV}
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	待扩展的NV索引 授权索引：None
TCM2B_MAX_NV_BUFFER	data	待扩展到NV区的数据

表 148　TCM2_NV_Extend()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.7　TCM2_NV_SetBits

该命令用于设置 NV 索引中的比特。任意数量从 0 到 64 的比特可被设置。NV 索引的内容从偏移量(offset)开始与 data 进行或运算。对 data 和 offset 的验证同 TCM2_NV_Write。

如果 TCMA_NV_WRITTEN 没有设置，NV 索引被认为是包含所有零比特，data 与该值进行或运算。

如果 TCMA_NV_BITS 没有设置，TCM 将会返回 TCM2_RC_ATTRIBUTES.

该命令成功完成后，NV 索引的 TCMA_NV_WRITTEN 属性将被设置。

该命令接口输入和输出具体参数要求见表 149 和表 150。

表 149　TCM2_NV_SetBits()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_SetBits {NV}

表 149 TCM2_NV_SetBits ( ) 接口输入参数 (续)

类型	名称	描述
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	需设置比特的NV区的NV索引 授权索引：None
UINT64	bits	与当前内容进行或操作的数据

表 150 TCM2_NV_SetBits( )接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.8 TCM2_NV_WriteLock

如果NV索引的TCMA_NV_WRITEDEFINE或TCMA_NV_WRITE_STCLEAR属性被设置，该命令用于抑制NV索引被再写。

该命令应使用正确的写授权。如果NV索引的TCMA_NV_WRITELOCKED属性已经被设置，该命令不会出错。

如果NV索引的TCMA_NV_WRITEDEFINE或TCMA_NV_WRITE_STCLEAR都没有设置，TCM将会返回TCM2_RC_ATTRIBUTES。

当命令使用了正确的授权，且TCMA_NV_WRITE_STCLEAR或TCMA_NV_WRITEDEFINE被设置，TCM将设置NV索引的TCMA_NV_WRITELOCKED。下一次TCM2_Startup(TCM2_SU_CLEAR)命令后TCMA_NV_WRITELOCKED将会被清除。

该命令接口输入和输出具体参数要求见表151和表152。

表 151 TCM2_NV_WriteLock( )接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_WriteLock {NV}
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	待锁定NV区的NV索引 授权索引：None

表 152 TCM2_NV_WriteLock()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.9 TCM2_NV_GlobalWriteLock

该命令为所有具有 TCMA_NV_GLOBALLOCK 属性的索引设置 TCMA_NV_WRITELOCKED 属性值。

如果索引具有 TCMA_NV_WRITELOCKED 和 TCMA_NV_WRITEDEFINE 属性,除了 TCMA_NV_WRITTEN 设置了 CLEAR 外,该命令将永久锁定该 NV 索引的写操作,授权为 platformAuth/platformPolicy 或者 ownerAuth/ownerPolicy。

该命令接口输入和输出具体参数要求见表 153 和表 154。

表 153 TCM2_NV_GlobalWriteLock()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_GlobalWriteLock
TCMI_RH_PROVISION	@authHandle	TCM2_RH_OWNER or TCM2_RH_PLATFORM+{PP} 授权索引:1 授权角色:USER

表 154 TCM2_NV_GlobalWriteLock()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出tag是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.10 TCM2_NV_Read

该命令用于往使用 TCM2_NV_DefineSpace()命令定义的 NV 空间读出值。

该命令的使用应使用正确的授权,如 TCMA_NV_PPREAD、TCMA_NV_OWNERREAD、TCMA_NV_AUTHREAD。

如果 NV 索引的 TCMA_NV_READLOCKE 属性设置,TCM 将会返回 TCM2_RC_NV_LOCKED。

该命令接口输入和输出具体参数要求见表 155 和表 156。

表 155　TCM2_NV_Read()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_Read
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	执行读操作的NV索引 授权索引：None
UINT16	size	准备读的字节数
UINT16	offset	NV区的偏移量 该值可小于或者等于nvIndex的数据大小

表 156　TCM2_NV_Read()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出tag是TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码
TCM2B_MAX_NV_BUFFER	data	读出的数据

7.21.11　TCM2_NV_ReadLock

如果索引的 TCMA_NV_READ_STCLEAR 属性设置，该命令用于抑制 NV 索引被再读，操作有效期到下一次 TCM2_Startup（TCM2_SU_CLEAR）。

该命令应使用正确的授权。如定义的 TCMA_NV_PPREAD、TCMA_NV_OWNERREAD 和 TCMA_NV_AUTHREAD。

当命令使用了正确的授权，且 TCMA_NV_READ_STCLEAR 被设置，TCM 将设置 NV 索引的 TCMA_NV_READLOCKED。如果该 NV 索引的 TCMA_NV_READ_STCLEAR 被清除，TCM 将返回 TCM2_RC_NV_ATTRIBUTE。下一次 TCM2_Startup（TCM2_SU_CLEAR）命令后 TCMA_NV_READLOCKED 将会被清除。

该命令接口输入和输出具体参数要求见表 157 和表 158。

表 157 TCM2_NV_ReadLock()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_ReadLock
TCMI_RH_NV_AUTH	@authHandle	指向授权值源的句柄 授权索引：1 授权角色：USER
TCMI_RH_NV_INDEX	nvIndex	待锁定的NV区的NV索引 授权索引：None

表 158 TCM2_NV_ReadLock()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功，输出标记应与输入标记相同，如果命令执行失败，则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

7.21.12 TCM2_NV_ChangeAuth

该命令允许NV索引的授权秘密被改变。如果成功，与nvIndex关联的授权秘密(authValue)会被改变。

该命令接口输入和输出具体参数要求见表159和表160。

表 159 TCM2_NV_ChangAuth()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	TCM2_ST_SESSIONS
UINT32	commandSize	命令长度
TCM2_CC	commandCode	TCM2_CC_NV_ChangeAuth
TCMI_RH_NV_INDEX	@nvIndex	对象句柄 授权索引：1 授权角色：ADMIN
TCM2B_AUTH	newAuth	新的授权值

表 160 TCM2_NV_ChangAuth()接口输出参数

类型	名称	描述
TCM2_ST	tag	如果命令执行成功,输出标记应与输入标记相同,如果命令执行失败,则输出 tag 是 TCM2_ST_NO_SESSIONS
UINT32	responseSize	返回数据大小
TCM2_RC	responseCode	返回码

8 可信密码模块证实方法

8.1 概述

本文件使用人员在应用和检测可信密码模块时应验证其是否符合第 7 章的要求,第 8 章是符合性实现的原理说明,更多关键元素的具体证实实例见附录 B。下述证实方法原理性描述以及附录 B 中的输入输出描述均采用了十六进制数字表示方法。

与 ISO/IEC 11889-3:2015 章条编号对照情况见附录 C。

典型应用接口的依赖关系见附录 D。

8.2 符合性实现原理说明

8.2.1 TCM2_Startup

本命令依赖于:复位命令的成功执行。具体接口参数见表 161 和表 162。

表 161 TCM2_Startup()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	Tag	80 01
UINT32	commandSize	00 00 00 0c
TCM2_CC	commandCode	00 00 01 44
TCM2_SU	startupType	00 00

表 162 TCM2_Startup()接口输出参数

类型	名称	描述
TCM2_ST	Tag	80 01
UINT32	responseSize	00 00 00 0a
TCM2_RC	responseCode	00 00 00 00

输入:

80010000000c000001440000

输出:

80010000000a00000000

8.2.2 TCM2_PCR_Extend

本命令依赖于：TCM2_Startup 命令的成功执行。具体接口参数见表 163 和表 164。

表 163 TCM2_PCR_Extend()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	80 02
UINT32	commandSize	00 00 00 41
TCM2_CC	commandCode	00 00 01 82
TCMI_DH_PCR+	@pcrHandle	00 00 00 00
TCML_DIGEST_VALUES	digests	00 00 00 01 00 12 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46

表 164 TCM2_PCR_Extend()接口输出参数

类型	名称	描述
TCM2_ST	tag	80 02
UINT32	responseSize	00 00 00 13
TCM2_RC	responseCode	00 00 00 00

输入：

8002000000410000018200000000000000094000000900000000000000000100123031323334353637383941424344454630313233343536373839414243444546

输出：

80020000001300000000000000000000010000

8.2.3 TCM2_PCR_Read

具体接口参数见表 165 和表 166。

表 165 TCM2_PCR_Read()接口输入参数

类型	名称	描述
TCMI_ST_COMMAND_TAG	tag	80 01
UINT32	commandSize	00 00 00 14
TCM2_CC	commandCode	00 00 01 7e
TCML_PCR_SELECTION	pcrSelectionIn	00 00 00 01 00 12 03 01 00 00

表 166　TCM2_PCR_Read()接口输出参数

类型	名称	描述
TCM2_ST	tag	80 01
UINT32	responseSize	00 00 00 3e
TCM2_RC	responseCode	00 00 00 00
UINT32	pcrUpdateCounter	00 00 00 15
TCML_PCR_SELECTION	pcrSelectionOut	00 00 00 01 00 12 03 01 00 00
TCML_DIGEST	pcrValues	00 00 00 01 00 20 09 29 e1 b5 6b 23 66 c9 79 00 01 04 96 dc 51 98 cc 00 40 01 e2 e6 dc ca f9 bb 98 cc 7e 50 24 56

输入：

8001000000140000017e00000001001203010000

输出：

80010000003e0000000000000015000000010012030100000000000100200929e1b56b2366c97900010496dc5198cc004001e2e6dccaf9bb98cc7e502456

附 录 A
（规范性）
数据结构

A.1 命令码

TCM2_CC 命令码的名称和值见表 A.1。

表 A.1 命令码

名称	命令码
reserved	0x0000011F
TCM2_CC_EvictControl	0x00000120
TCM2_CC_HierarchyControl	0x00000121
TCM2_CC_NV_UndefineSpace	0x00000122
reserved	0x00000124
reserved	0x00000125
TCM2_CC_Clear	0x00000126
TCM2_CC_ClearControl	0x00000127
Reserved	0x00000128
TCM2_CC_HierarchyChangeAuth	0x00000129
TCM2_CC_NV_DefineSpace	0x0000012A
reserved	0x0000012B
reserved	0x0000012C
TCM2_CC_PP_Commands	0x0000012D
TCM2_CC_SetPrimaryPolicy	0x0000012E
reserved	0x0000012F
reserved	0x00000130
TCM2_CC_CreatePrimary	0x00000131
TCM2_CC_NV_GlobalWriteLock	0x00000132
reserved	0x00000133
TCM2_CC_NV_Increment	0x00000134
TCM2_CC_NV_SetBits	0x00000135
TCM2_CC_NV_Extend	0x00000136
TCM2_CC_NV_Write	0x00000137
TCM2_CC_NV_WriteLock	0x00000138
TCM2_CC_DictionaryAttackLockReset	0x00000139
TCM2_CC_DictionaryAttackParameters	0x0000013A

表 A.1 命令码（续）

名称	命令码
TCM2_CC_NV_ChangeAuth	0x0000013B
reserved	0x0000013C
TCM2_CC_PCR_Reset	0x0000013D
TCM2_SequenceComplete	0x0000013E
reserved	0x0000013F
reserved	0x00000140
reserved	0x00000141
TCM2_CC_IncrementalSelfTest	0x00000142
TCM2_CC_SelfTest	0x00000143
TCM2_CC_Startup	0x00000144
TCM2_CC_Shutdown	0x00000145
reserved	0x00000146
TCM2_CC_ActivateCredential	0x00000147
TCM2_CC_Certify	0x00000148
reserved	0x00000149
TCM2_CC_CertifyCreation	0x0000014A
TCM2_CC_Duplicate	0x0000014B
TCM2_CC_GetTime	0x0000014C
reserved	0x0000014D
TCM2_CC_NV_Read	0x0000014E
TCM2_CC_NV_ReadLock	0x0000014F
TCM2_CC_ObjectChangeAuth	0x00000150
TCM2_CC_PolicySecret	0x00000151
TCM2_CC_Rewrap	0x00000152
TCM2_CC_Create	0x00000153
TCM2_CC_ECDH_ZGen	0x00000154
TCM2_CC_HMAC	0x00000155
TCM2_CC_Import	0x00000156
TCM2_CC_Load	0x00000157
TCM2_CC_Quote	0x00000158
reserved	0x00000159
TCM2_CC_HMAC_Start	0x0000015B
TCM2_CC_SequenceUpdate	0x0000015C
TCM2_CC_Sign	0x0000015D

表 A.1 命令码（续）

名称	命令码
TCM2_CC_Unseal	0x0000015E
TCM2_CC_PolicySigned	0x00000160
TCM2_CC_ContextLoad	0x00000161
TCM2_CC_ContextSave	0x00000162
TCM2_CC_ECDH_KeyGen	0x00000163
reserved	0x00000164
TCM2_CC_FlushContext	0x00000165
reserved	0x00000166
TCM2_CC_LoadExternal	0x00000167
TCM2_CC_MakeCredential	0x00000168
TCM2_CC_NV_ReadPublic	0x00000169
reserved	0x0000016A
TCM2_CC_PolicyAuthValue	0x0000016B
TCM2_CC_PolicyCommandCode	0x0000016C
TCM2_CC_PolicyCpHash	0x0000016E
reserved	0x0000016F
TCM2_CC_PolicyOR	0x00000171
TCM2_CC_PolicyTicket	0x00000172
TCM2_CC_ReadPublic	0x00000173
reserved	0x00000174
TCM2_CC_StartAuthSession	0x00000176
TCM2_CC_VerifySignature	0x00000177
reserved	0x00000178
reserved	0x00000179
TCM2_CC_GetCapability	0x0000017A
TCM2_CC_GetRandom	0x0000017B
TCM2_CC_GetTestResult	0x0000017C
TCM2_CC_Hash	0x0000017D
TCM2_CC_PCR_Read	0x0000017E
TCM2_CC_PolicyPCR	0x0000017F
TCM2_CC_PolicyRestart	0x00000180
reserved	0x00000181
TCM2_CC_PCR_Extend	0x00000182
reserved	0x00000183

表 A.1 命令码（续）

名称	命令码
reserved	0x00000184
reserved	0x00000185
TCM2_CC_HashSequenceStart	0x00000186
TCM2_CC_PolicyPhysicalPresence	0x00000187
reserved	0x00000188
TCM2_CC_PolicyGetDigest	0x00000189
TCM2_CC_TestParms	0x0000018A
TCM2_CC_Commit	0x0000018B
TCM2_CC_PolicyPassword	0x0000018C
TCM2_CC_ZGen_2Phase	0x0000018D
TCM2_CC_EC_Ephemeral	0x0000018E
reserved	0x0000018F
reserved	0x00000190
reserved	0x00000191
reserved	0x00000192
TCM2_CC_EncryptDecrypt	0x00000193
reserved	0x00000194
reserved	0x00000195
reserved	0x00000196
reserved	0x00000197
reserved	0x00000198
TCM2_CC_ECC_Encrypt	0x00000199
TCM2_CC_ECC_Decrypt	0x0000019A
TCM2_CC_LAST	0x0000019A
CC_VEND	0x20000000
注：表中“reserved”指用于标准兼容性而保留使用的数据字段或操作。	

A.2 返回码

返回码请见表 A.2。

表 A.2 返回码

名称	值
TCM2_RC_SUCCESS	0x000
TCM2_RC_BAD_TAG	0x01E

表 A.2 返回码(续)

名称	值
RC_VER1	0x100
TCM2_RC_INITIALIZE	RC_VER1+0x000
TCM2_RC_FAILURE	RC_VER1+0x001
TCM2_RC_SEQUENCE	RC_VER1+0x003
TCM2_RC_PRIVATE	RC_VER1+0x00B
TCM2_RC_HMAC	RC_VER1+0x019
TCM2_RC_DISABLED	RC_VER1+0x020
TCM2_RC_EXCLUSIVE	RC_VER1+0x021
TCM2_RC_AUTH_TYPE	RC_VER1+0x024
TCM2_RC_AUTH_MISSING	RC_VER1+0x025
TCM2_RC_POLICY	RC_VER1+0x026
TCM2_RC_PCR	RC_VER1+0x027
TCM2_RC_PCR_CHANGED	RC_VER1+0x028
TCM2_RC_UPGRADE	RC_VER1+0x02D
TCM2_RC_TOO_MANY_CONTEXTS	RC_VER1+0x02E
TCM2_RC_AUTH_UNAVAILABLE	RC_VER1+0x02F
TCM2_RC_REBOOT	RC_VER1+0x030
TCM2_RC_UNBALANCED	RC_VER1+0x031
TCM2_RC_COMMAND_SIZE	RC_VER1+0x042
TCM2_RC_COMMAND_CODE	RC_VER1+0x043
TCM2_RC_AUTHSIZE	RC_VER1+0x044
TCM2_RC_AUTH_CONTEXT	RC_VER1+0x045
TCM2_RC_NV_RANGE	RC_VER1+0x046
TCM2_RC_NV_SIZE	RC_VER1+0x047
TCM2_RC_NV_LOCKED	RC_VER1+0x048
TCM2_RC_NV_AUTHORIZATION	RC_VER1+0x049
TCM2_RC_NV_UNINITIALIZED	RC_VER1+0x04A
TCM2_RC_NV_SPACE	RC_VER1+0x04B
TCM2_RC_NV_DEFINED	RC_VER1+0x04C
TCM2_RC_BAD_CONTEXT	RC_VER1+0x050
TCM2_RC_CPHASH	RC_VER1+0x051
TCM2_RC_PARENT	RC_VER1+0x052
TCM2_RC_NEEDS_TEST	RC_VER1+0x053
TCM2_RC_NO_RESULT	RC_VER1+0x054

表 A.2 返回码(续)

名称	值
TCM2_RC_SENSITIVE	RC_VER1+0x055
RC_MAX_FM0	RC_VER1+0x07F
RC_FMT1	0x080
TCM2_RC_ASYMMETRIC	RC_FMT1+0x001
TCM2_RC_ATTRIBUTES	RC_FMT1+0x002
TCM2_RC_HASH	RC_FMT1+0x003
TCM2_RC_VALUE	RC_FMT1+0x004
TCM2_RC_HIERARCHY	RC_FMT1+0x005
TCM2_RC_KEY_SIZE	RC_FMT1+0x007
TCM2_RC_MGF	RC_FMT1+0x008
TCM2_RC_MODE	RC_FMT1+0x009
TCM2_RC_TYPE	RC_FMT1+0x00A
TCM2_RC_HANDLE	RC_FMT1+0x00B
TCM2_RC_KDF	RC_FMT1+0x00C
TCM2_RC_RANGE	RC_FMT1+0x00D
TCM2_RC_AUTH_FAIL	RC_FMT1+0x00E
TCM2_RC_NONCE	RC_FMT1+0x00F
TCM2_RC_PP	RC_FMT1+0x010
TCM2_RC_SCHEME	RC_FMT1+0x012
TCM2_RC_SIZE	RC_FMT1+0x015
TCM2_RC_SYMMETRIC	RC_FMT1+0x016
TCM2_RC_TAG	RC_FMT1+0x017
TCM2_RC_SELECTOR	RC_FMT1+0x018
TCM2_RC_INSUFFICIENT	RC_FMT1+0x01A
TCM2_RC_SIGNATURE	RC_FMT1+0x01B
TCM2_RC_KEY	RC_FMT1+0x01C
TCM2_RC_POLICY_FAIL	RC_FMT1+0x01D
TCM2_RC_INTEGRITY	RC_FMT1+0x01F
TCM2_RC_TICKET	RC_FMT1+0x020
TCM2_RC_RESERVED_BITS	RC_FMT1+0x021
TCM2_RC_BAD_AUTH	RC_FMT1+0x022
TCM2_RC_EXPIRED	RC_FMT1+0x023
TCM2_RC_POLICY_CC	RC_FMT1+0x024
TCM2_RC_BINDING	RC_FMT1+0x025

表 A.2 返回码(续)

名称	值
TCM2_RC_CURVE	RC_FMT1+0x026
TCM2_RC_ECC_POINT	RC_FMT1+0x027
RC_WARN	0x900
TCM2_RC_CONTEXT_GAP	RC_WARN+0x001
TCM2_RC_OBJECT_MEMORY	RC_WARN+0x002
TCM2_RC_SESSION_MEMORY	RC_WARN+0x003
TCM2_RC_MEMORY	RC_WARN+0x004
TCM2_RC_SESSION_HANDLES	RC_WARN+0x005
TCM2_RC_OBJECT_HANDLES	RC_WARN+0x006
TCM2_RC_LOCALITY	RC_WARN+0x007
TCM2_RC_YIELDED	RC_WARN+0x008
TCM2_RC_CANCELED	RC_WARN+0x009
TCM2_RC_TESTING	RC_WARN+0x00A
TCM2_RC_REFERENCE_H0	RC_WARN+0x010
TCM2_RC_REFERENCE_H1	RC_WARN+0x011
TCM2_RC_REFERENCE_H2	RC_WARN+0x012
TCM2_RC_REFERENCE_H3	RC_WARN+0x013
TCM2_RC_REFERENCE_H4	RC_WARN+0x014
TCM2_RC_REFERENCE_H5	RC_WARN+0x015
TCM2_RC_REFERENCE_H6	RC_WARN+0x016
TCM2_RC_REFERENCE_S0	RC_WARN+0x018
TCM2_RC_REFERENCE_S1	RC_WARN+0x019
TCM2_RC_REFERENCE_S2	RC_WARN+0x01A
TCM2_RC_REFERENCE_S3	RC_WARN+0x01B
TCM2_RC_REFERENCE_S4	RC_WARN+0x01C
TCM2_RC_REFERENCE_S5	RC_WARN+0x01D
TCM2_RC_REFERENCE_S6	RC_WARN+0x01E
TCM2_RC_NV_RATE	RC_WARN+0x020
TCM2_RC_LOCKOUT	RC_WARN+0x021
TCM2_RC_RETRY	RC_WARN+0x022
TCM2_RC_NV_UNAVAILABLE	RC_WARN+0x023
TCM2_RC_NOT_USED	RC_WARN+0x7F
TCM2_RC_H	0x000
TCM2_RC_P	0x040

表 A.2 返回码(续)

名称	值
TCM2_RC_S	0x800
TCM2_RC_1	0x100
TCM2_RC_2	0x200
TCM2_RC_3	0x300
TCM2_RC_4	0x400
TCM2_RC_5	0x500
TCM2_RC_6	0x600
TCM2_RC_7	0x700
TCM2_RC_8	0x800
TCM2_RC_9	0x900
TCM2_RC_A	0xA00
TCM2_RC_B	0xB00
TCM2_RC_C	0xC00
TCM2_RC_D	0xD00
TCM2_RC_E	0xE00
TCM2_RC_F	0xF00
TCM2_RC_N_MASK	0xF00

A.3 基本常量

A.3.1 基本类型

基本类型的定义见表 A.3。

表 A.3 基本类型定义

类型	名称	描述
uint8_t	UINT8	无符号,8 比特整型
uint8_t	BYTE	无符号,8 比特整型
int8_t	INT8	有符号,8 比特整型
int	BOOL	1 比特整型
uint16_t	UINT16	无符号,16 比特整型
int16_t	INT16	有符号,16 比特整型
uint32_t	UINT32	无符号,32 比特整型
int32_t	INT32	有符号,32 比特整型
uint64_t	UINT64	无符号,64 比特整型
int64_t	INT64	有符号,32 比特整型

A.3.2 逻辑数值常量

逻辑数值常量的定义见表 A.4。

表 A.4 逻辑数值定义

名称	值	描述
TRUE	1	—
FALSE	0	—
YES	1	—
NO	0	—
SET	1	—
CLEAR	0	—

A.3.3 其他类型

其他类型的定义见表 A.5。

表 A.5 其他类型

类型	名称	描述
UINT32	TCM2_ALGORITHM_ID	兼容上一版本
UINT32	TCM2_MODIFIER_INDICATOR	
UINT32	TCM2_AUTHORIZATION_SIZE	命令中的 authorizationSize 参数
UINT32	TCM2_PARAMETER_SIZE	命令中的 parameterSize 参数
UINT16	TCM2_KEY_SIZE	密钥大小(10进制)
UINT16	TCM2_KEY_BITS	密钥大小(2进制)

A.3.4 常量

A.3.4.1 TCM2_SPEC(标准版本值)

TCM2_SPEC 常量定义见表 A.6,定义的值由 TCM2_GetCapability()读出。

表 A.6 TCM2_SPEC 常量定义

类型	值	描述
TCM2_SPEC_FAMILY	0x322E3000	ASCII“2.0”,NULL结尾
reserved	—	—
reserved	—	—
TCM2_SPEC_YEAR	2022	年份版本
reserved	—	—
注:表中“reserved”指用于标准兼容性而保留使用的数据字段或者操作。		

A.3.4.2 TCM2_ALG_ID

TCM2_ALG_ID 图例及 ID 的定义见表 A.7 到表 A.9。

注 1：在类型列中名称前面用“+”表示扩展值，表示这是在某些情况下允许的条件值。

注 2：验证代码确定输入参数的类型错误，则该表可能包含一个以“=”字符为前缀的条目，以指示响应代码。

注 3：表中“reserved”指用于标准兼容性而保留使用的数据字段或操作。

注 4：附录 A 表中出现“+”、“=”、“reserved”内容，有注的，参照各表中的注，没有注的，参照本注 1、2、3 解释内容。

表 A.7 TCM2_ALG_ID 表图例

列名称	描述
算法名称	分配给算法的助记名称
值	分配给算法的值
类型	允许的值如下： A——含有公钥和私钥的非对称密码算法 S——仅有一个私钥的对称密码算法 H——将输入数据压缩成摘要值的杂凑算法 X——签名密码算法 N——一个匿名签名算法 E——一个加密算法 M——一个如 Mask 产生功能的方法 O——一个对象类型
依赖	如果这个算法被声明，表明还应声明其他算法
参考	定义算法参考的文档
描述	说明

表 A.8 TCM2_ALG_ID(UINT16)定义

算法名称	值	类型	依赖	描述
TCM2_ALG_ERROR	0x0000	—	—	不应发生
reserved	0x0001	—	—	—
reserved	0x0004	—	—	—
TCM2_ALG_HMAC	0x0005	H X	—	HMAC 算法
reserved	0x0006～0x0007	—	—	—
reserved	0x0008	—	—	—
reserved	0x0009	—	—	—
reserved	0x000A	—	—	—
reserved	0x000B～0X000F	—	—	—
TCM2_ALG_NULL	0x0010	—	—	—
TCM2_ALG_SM3_256	0x0012	H	—	SM3 杂凑算法
TCM2_ALG_SM4	0x0013	S	—	SM4 对称算法
reserved	0x0014～0x0019	—	—	—

表 A.8　TCM2_ALG_ID（UINT16）定义（续）

算法名称	值	类型	依赖	描述
TCM2_ALG_ECDAA	0x001A	A X N	ECC	
TCM2_ALG_SM2	0x001B	A X E	—	SM2非对称算法
reserved	0x001C～0x001F	—	—	—
TCM2_ALG_KDF1_SP800_56A	0x0020	H M	ECC	
TCM2_ALG_KDF2	0x0021	H M		
TCM2_ALG_KDF1_SP800_108	0x0022	H M		
TCM2_ALG_ECC	0x0023	A O		
reserved	0　X0024		—	—
TCM2_ALG_SYMCIPHER	0x0025	O S		
reserved	0x0026～0x0042		—	—
TCM2_ALG_CFB	0x0043	S E	—	—
reserved	0x0044		—	—
reserved	0x00C1　～0x00C6		—	保留
reserved	0x8000～0xFFFF		—	保留
注：表中“reserved”指用于标准兼容性而保留使用的数据字段或者操作。				

表 A.9　TCM2_ECC_CURVE(UINT16)定义

名称	值	描述
reserved	0x0000	—
reserved	0x0003	—
reserved	0x0004	—
TCM2_ECC_SM2_P256	0x0020	—
≠TCM2_RC_CURVE	—	—
注：表中“reserved”指用于标准兼容性而保留使用的数据字段或者操作。		

A.3.4.3　TCM2_CC（命令码）

一个命令是一个 32 位构，其字段被分配，见表 A.10 和表 A.11。

表 A.10　TCM 命令 32 位结构

31	30	29	28												16	15															0
res		V	reserved													命令索引															

表 A.11 TCM 命令格式字段描述

比特	名称	定义
15:0	Command Index	命令索引
28:16	reserved	应为 0
29	V	SET(1):厂商自定义命令 CLEAR(0):非厂商自定义的命令
31:30	res	应为 0

A.3.4.4 TCM2_ST(会话类型)

TCM2_ST 的定义见表 A.12。

表 A.12 TCM2_ST(UINT16)定义

名称	值	描述
TCM2_ST_RSP_COMMAND	0x00C4	返回值的标识 用于早期版本 TCM 规范的命令的返回码
TCM2_ST_NULL	0X8000	保留,未定义具体含义
TCM2_ST_NO_SESSIONS	0x8001	命令/返回的标识的值 应用的命令/返回值不涉及会话,无授权大小/参数 如果 TCM 的返回码(responseCode)不是 TCM2_SUCCESS,输出参数的 tag 应是该值
TCM2_ST_SESSIONS	0x8002	命令/返回的标识的值 表明命令/返回有一个或者多个会话,授权值/参数大小值
reserved	0x8003	保留
reserved	0x8004	保留
TCM2_ST_ATTEST_NV	0x8014	标识用于认证的结构体
reserved	0x8015	保留
reserved	0x8016	保留
TCM2_ST_ATTEST_CERTIFY	0x8017	标识用于认证的结构体
TCM2_ST_ATTEST_QUOTE	0x8018	标识用于认证的结构体
TCM2_ST_ATTEST_TIME	0x8019	标识用于认证的结构体
TCM2_ST_ATTEST_CREATION	0x801A	标识用于认证的结构体
reserved	0x801B	保留
TCM2_ST_CREATION	0x8021	标识用于 ticket 类型
TCM2_ST_VERIFIED	0x8022	标识用于 ticket 类型
TCM2_ST_AUTH_SECRET	0x8023	标识用于 ticket 类型
TCM2_ST_HASHCHECK	0x8024	标识用于 ticket 类型
TCM2_ST_AUTH_SIGNED	0x8025	标识用于 ticket 类型
注:表中“reserved”指用于标准兼容性而保留使用的数据字段或者操作。		

A.3.4.5 TCM2_CAP（Capilities）

TCM2_CAP 的定义见表 A.13。

表 A.13 TCM2_CAP 定义

名称	值	属性	返回类型
TCM2_CAP_FIRST	0x00000000		
TCM2_CAP_ALGS	0x00000000	TCM2_ALG_ID(1)	TCML_ALG_PROPERTY
TCM2_CAP_HANDLES	0x00000001	TCM2_HANDLE	TCML_HANDLE
TCM2_CAP_COMMANDS	0x00000002	TCM2_CC	TCML_CC
TCM2_CAP_PP_COMMANDS	0x00000003	TCM2_CC	TCML_CC
reserved	0x00000004	TCM2_CC	TCML_CC
TCM2_CAP_PCRS	0x00000005	reserved	TCML_PCR_SELECTION
TCM2_CAP_TCM2_PROPERTIES	0x00000006	TCM2_PT	TCML_TAGGED_TCM2_PROPERTY
TCM2_CAP_PCR_PROPERTIES	0x00000007	TCM2_PT_PCR	TCML_TAGGED_PCR_PROPERTY
TCM2_CAP_ECC_CURVES	0x00000008	TCM2_ECC_CURVE(1)	TCML_ECC_CURVE
TCM2_CAP_LAST	0x00000008		
TCM2_CAP_VENDOR_PROPERTY	0x00000100	厂商自定义	厂商自定义的值
≠TCM2_RC_VALUE			

注：TCM2_ALG_ID或者 TCM2_ECC_CURVE为 UINT32类型。

A.3.4.6 TCM2_PT（属性标识）

TCM2_PT 的定义见表 A.14。

表 A.14 TCM2_PT 定义

名称	值	描述
TCM2_PT_NONE	0x00000000	表明无属性类型
PT_GROUP	0x00000100	每一个组的属性个数
PT_FIXED	PT_GROUP * 1	以 TCMS_TAGGED_PROPERTY 返回组中固定属性仅在TCM固件更改时，组中的这个值才会改变
TCM2_PT_FAMILY_INDICATOR	PT_FIXED＋0	一个含有 TCM 族的4字节字符串
TCM2_PT_LEVEL	PT_FIXED＋1	本文件的level为0
TCM2_PT_REVISION	PT_FIXED＋2	规范修订的版本号乘以100 例如，修订版本号为01.01，结果会是101
TCM2_PT_DAY_OF_YEAR	PT_FIXED＋3	规范的日期 例如：11/15/2010为一年中的第319天，值为319（00 00 01 3F）

表 A.14 TCM2_PT 定义（续）

名称	值	描述
TCM2_PT_YEAR	PT_FIXED+4	规范使用的年份 例如： 2010 年为00 00 07 DA
TCM2_PT_MANUFACTURER	PT_FIXED+5	每个厂商的统一ID
TCM2_PT_VENDOR_STRING_1	PT_FIXED+6	厂商ID的前四个字符
TCM2_PT_VENDOR_STRING_2	PT_FIXED+7	厂商ID的第二个四字符
TCM2_PT_VENDOR_STRING_3	PT_FIXED+8	厂商ID的第三个四字符
TCM2_PT_VENDOR_STRING_4	PT_FIXED+9	厂商ID的第四个四字符
TCM2_PT_VENDOR_TCM2_TYPE	PT_FIXED+10	厂商自定义的表明TCM模式的
TCM2_PT_FIRMWARE_VERSION_1	PT_FIXED+11	高位32比特,用于定义厂商固件版本号
TCM2_PT_FIRMWARE_VERSION_2	PT_FIXED+12	低位32比特,用于定义厂商固件版本号
TCM2_PT_INPUT_BUFFER	PT_FIXED+13	参数的最大值(典型的为 TCM2B_MAX_BUFFER=1024)
TCM2_PT_HR_TRANSIENT_MIN	PT_FIXED+14	在TCM RAM中可容纳临时对象的最小值
TCM2_PT_HR_PERSISTENT_MIN	PT_FIXED+15	在TCM NV内存中可容纳永久对象的最小值
TCM2_PT_HR_LOADED_MIN	PT_FIXED+16	在TCM RAM中可容纳授权会话的最小值
TCM2_PT_ACTIVE_SESSIONS_MAX	PT_FIXED+17	一次可激活的授权会话的数量
TCM2_PT_PCR_COUNT	PT_FIXED+18	PCR声明的个数
TCM2_PT_PCR_SELECT_MIN	PT_FIXED+19	在TCMS_PCR_SELECT.sizeOfSelect中的8字节最小值
TCM2_PT_CONTEXT_GAP_MAX	PT_FIXED+20	两个最大保存的上下文会话ID的允许最大值 至少为 $2^{16}-1$ (65535)
	PT_FIXED+21	保留
TCM2_PT_NV_COUNTERS_MAX	PT_FIXED+22	被允许设置TCMA_NV_COUNTER的最大NV索引值
TCM2_PT_NV_INDEX_MAX	PT_FIXED+23	最大NV索引数据区域
TCM2_PT_MEMORY	PT_FIXED+24	用于TCM管理方法的内存
TCM2_PT_CLOCK_UPDATE	PT_FIXED+25	内部的,毫秒(ms)
TCM2_PT_CONTEXT_HASH	PT_FIXED+26	该算法在在上下文保存用完整的HMAC
TCM2_PT_CONTEXT_SYM	PT_FIXED+27	算法用于保存上下文的加密
TCM2_PT_CONTEXT_SYM_SIZE	PT_FIXED+28	用于上下文加密的密钥的大小
TCM2_PT_ORDERLY_COUNT	PT_FIXED+29	NV有序计数器的大小
TCM2_PT_MAX_COMMAND_SIZE	PT_FIXED+30	命令发送TCM最大长度值
TCM2_PT_MAX_RESPONSE_SIZE	PT_FIXED+31	命令TCM返回数据最大长度值
TCM2_PT_MAX_DIGEST	PT_FIXED+32	生成摘要的最大长度
TCM2_PT_MAX_OBJECT_CONTEXT	PT_FIXED+33	上下文对象占用的最大空间

表 A.14 TCM2_PT 定义（续）

名称	值	描述
TCM2_PT_MAX_SESSION_CONTEXT	PT_FIXED+34	上下文会话占用的最大空间
TCM2_PT_PS_FAMILY_INDICATOR	PT_FIXED+35	标准版本号
TCM2_PT_PS_LEVEL	PT_FIXED+36	标准层次
TCM2_PT_PS_REVISION	PT_FIXED+37	标准平台号
TCM2_PT_PS_DAY_OF_YEAR	PT_FIXED+38	标准发布日期
TCM2_PT_PS_YEAR	PT_FIXED+39	标准发布年份
TCM2_PT_SPLIT_MAX	PT_FIXED+40	支持的最大拆分签名值
TCM2_PT_TOTAL_COMMANDS	PT_FIXED+41	支持的全部命令个数
TCM2_PT_LIBRARY_COMMANDS	PT_FIXED+42	已实现的支持命令个数
TCM2_PT_VENDOR_COMMANDS	PT_FIXED+43	厂商自定义命令个数
TCM2_PT_NV_BUFFER_MAX	PT_FIXED+44	非易失性存储是最大空间
TCM2_PT_MODES	PT_FIXED+45	运行模式
TCM2_PT_MAX_CAP_BUFFER	PT_FIXED+46	可查询属性最大空间
PT_VAR	PT_GROUP * 2	以TCMS_TAGGED_PROPERTY返回组中可变属性
TCM2_PT_PERMANENT	PT_VAR+0	TCMA_PERMANENT
TCM2_PT_STARTUP_CLEAR	PT_VAR+1	TCMA_STARTUP_CLEAR
TCM2_PT_HR_NV_INDEX	PT_VAR+2	已定义的NV索引数量
TCM2_PT_HR_LOADED	PT_VAR+3	已经加载到TCM RAM里的授权会话数量
TCM2_PT_HR_LOADED_AVAIL	PT_VAR+4	除了授权会话外，已经加载到TCM RAM里的任何类型会话的数量
TCM2_PT_HR_ACTIVE	PT_VAR+5	TCM当前跟踪的活动授权会话数量 为已加载的已保存的会话的总和
TCM2_PT_HR_ACTIVE_AVAIL	PT_VAR+6	除了授权会话外，可以创建的任何类型的会话
TCM2_PT_HR_TRANSIENT_AVAIL	PT_VAR+7	可以加载到TCM RAM中的其他瞬态对象的数量
TCM2_PT_HR_PERSISTENT	PT_VAR+8	当前加载到TCM NV内存中的持久对象数
TCM2_PT_HR_PERSISTENT_AVAIL	PT_VAR+9	可以加载到NV内存中的其他持久对象的数量
TCM2_PT_NV_COUNTERS	PT_VAR+10	具有NV TCMA_NV_COUNTER属性SET的已定义NV索引的数量
TCM2_PT_NV_COUNTERS_AVAIL	PT_VAR+11	可以使用其TCMA_NV_COUNTER和TCMA_NV_ORDERLY属性SET定义的附加NV索引的数量
TCM2_PT_ALGORITHM_SET	PT_VAR+12	限制了TCM可能使用的算法
TCM2_PT_LOADED_CURVES	PT_VAR+13	已经加载的ECC椭圆曲线的数量
TCM2_PT_LOCKOUT_COUNTER	PT_VAR+14	锁定计数器的当前值(failedTries)
TCM2_PT_MAX_AUTH_FAIL	PT_VAR+15	调用DA锁定之前的授权失败次数

表 A.14 TCM2_PT 定义（续）

名称	值	描述
TCM2_PT_LOCKOUT_INTERVAL	PT_VAR+16	TCM2_PT_LOCKOUT_COUNTER报告的值递减前的秒数
TCM2_PT_LOCKOUT_RECOVERY	PT_VAR+17	lockoutAuth失败后再次尝试使用lockoutAuth之前的秒数
TCM2_PT_NV_WRITE_RECOVERY	PT_VAR+18	TCM接受另一个修改NV的命令之前的毫秒数
reserved	PT_VAR+19	
reserved	PT_VAR+20	

A.3.4.7 TCM2_PT_PCR（PCR属性标识）

TCM2_PT_PCR的定义见表A.15。

表 A.15 TCM2_PT_PCR(UINT32)定义

名称	值	描述
TCM2_PT_PCR_FIRST	0x00000000	TCM2_PT_PCR属性范围的最底部
TCM2_PT_PCR_SAVE	0x00000000	TCMS_PCR_SELECT一个比特被设置，用于表明PCR被TCM2_SU_STATE保存和恢复
TCM2_PT_PCR_EXTEND_L0	0x00000001	TCMS_PCR_SELECT一个比特被设置，表明PCR从locality 0扩展
TCM2_PT_PCR_RESET_L0	0x00000002	TCMS_PCR_SELECT一个比特被设置，表明PCR是被TCM2_PCR_Reset()从locality 0被重置
reserved	0x00000003～0x00000010	保留
TCM2_PT_PCR_NO_INCREMENT	0x00000011	TCMS_PCR_SELECT一个比特被设置，表明修改PCR（重置或者扩展）将不会使pcrUpdateCounter增加
reserved	0x00000012	保留
TCM2_PT_PCR_POLICY	0x00000013	TCMS_PCR_SELECT一个比特被设置，表明PCR是被策略控制的
TCM2_PT_PCR_AUTH	0x00000014	TCMS_PCR_SELECT一个比特被设置，表明PCR是被授权值控制的
reserved	0x00000015	保留
reserved	0x00000016	保留
reserved	0x00000017～0x00000210	保留
reserved	0x00000211	保留
reserved	0x00000212	保留
reserved	0x00000213	被分配的PCR的属性值的初始值
TCM2_PT_PCR_LAST	0x00000014	声明的TCM2_PT_PCR属性范围的最大值

A.3.5 句柄

A.3.5.1 TCM2_HT(句柄类型)

句柄的定义见表 A.16,常量的定义见表 A.17。

表 A.16 TCM2_HT 句柄定义

类型	名称	描述
UINT32	TCM2_HANDLE	

表 A.17 TCM2_HT(UINT8)常量定义

名称	值	描述
TCM2_HT_PCR	0x00	PCR连续的数字,从0开始,涉及到PCR寄存器 一个平台设置一个最小的PCR数量,可声明更多
TCM2_HT_NV_INDEX	0x01	NV Index——分配给调用者
TCM2_HT_HMAC_SESSION	0x02	HMAC 授权会话——当创建了会话后分配给TCM
TCM2_HT_LOADED_SESSION	0x02	加载授权会话——仅在 TCM2_GetCapability上下文中使用 涉及到加载HMAC和加载粗略授权会话
TCM2_HT_POLICY_SESSION	0x03	策略授权会话——当创建了会话后分配给TCM
TCM2_HT_ACTIVE_SESSION	0x03	激活授权会话——仅在 TCM2_GetCapability上下文中使用 这个涉及到为TCM正在维持追踪信息的时候保存授权会话上下文
TCM2_HT_PERMANENT	0x40	永久值
TCM2_HT_TRANSIENT	0x80	临时对象——当一个对象被加载到临时对象内存中或者持续对象被转换成一个临时对象的时候由TCM分配
TCM2_HT_PERSISTENT	0x81	持续对象——当加载临时对象持久化时由TCM分配

A.3.5.2 TCM2_RH(永久句柄)

永久句柄的定义见表 A.18。

表 A.18 TCM2_RH(TCM2_HANDLE)定义

名称	值	类型	描述
TCM2_RH_FIRST	0x40000000	R	
TCM2_RH_SRK	0x40000000	R	保留
TCM2_RH_OWNER	0x40000001	K, A, P	用于 Storage Primary Seed (SPS), ownerAuth, 和 ownerPolicy 的 handle
TCM2_RH_REVOKE	0x40000002	R	保留
TCM2_RH_TRANSPORT	0x40000003	R	保留
TCM2_RH_OPERATOR	0x40000004	R	保留
TCM2_RH_ADMIN	0x40000005	R	保留

表 A.18 TCM2_RH（TCM2_HANDLE)定义（续）

名称	值	类型	描述
TCM2_RH_EK	0x40000006	R	保留
TCM2_RH_NULL	0x40000007	K，A，P	一个关联空层级的句柄
TCM2_RH_UNASSIGNED	0x40000008	R	保留
TCM2_RS_PW	0x40000009	S	用于声明密码授权会话的授权值
TCM2_RH_LOCKOUT	0x4000000A	A	用于字典攻击的lockout的重置授权
TCM2_RH_ENDORSEMENT	0x4000000B	K，A，P	用于背书原始种(Endorsement Primary Seed，EPS)，背书授权(endorsementAuth) 和 背书策略(endorsementPolicy)
TCM2_RH_PLATFORM	0x4000000C	K，A，P	用于平台原始种子(Platform Primary Seed，PPS)，平台授权(platformAuth)和平台策略(platformPolicy)
TCM2_RH_PLATFORM_NV	0x4000000D	C	用于phEnableNV
TCM2_RH_AUTH_00	0x40000010	A	厂商自定义的一个授权范围值的起始值
TCM2_RH_AUTH_FF	0x4000010F	A	厂商自定义的一个授权范围值的最大值
TCM2_RH_LAST	0x4000010F	R	保留
类型定义： R——保留值； K——原始种子； A——授权值； P——策略值； S——会话handle； C——控制			

A.3.5.3 TCM2_HC（Handle值常量）

TCM2_RH(TCM2_HANDLE)常量的定义见表 A.19。

表 A.19 TCM2_RH(TCM2_HANDLE)常量定义

名称	值	说明
HR_HANDLE_MASK	0x00FFFFFF	
HR_RANGE_MASK	0xFF000000	
HR_SHIFT	24	
HR_PCR	(TCM2_HT_PCR << HR_SHIFT)	
HR_HMAC_SESSION	(TCM2_HT_HMAC_SESSION << HR_SHIFT)	
HR_POLICY_SESSION	(TCM2_HT_POLICY_SESSION << HR_SHIFT)	
HR_TRANSIENT	(TCM2_HT_TRANSIENT << HR_SHIFT)	

表 A.19 TCM2_RH（TCM2_HANDLE）常量定义（续）

名称	值	说明
HR_PERSISTENT	(TCM2_HT_PERSISTENT << HR_SHIFT)	
HR_NV_INDEX	(TCM2_HT_NV_INDEX << HR_SHIFT)	
HR_PERMANENT	(TCM2_HT_PERMANENT << HR_SHIFT)	
PCR_FIRST	(HR_PCR+0)	第一个PCR
PCR_LAST	(PCR_FIRST+IMPLEMENTATION_PCR−1)	最后一个PCR
HMAC_SESSION_FIRST	(HR_HMAC_SESSION+0)	第一个HMAC session
HMAC_SESSION_LAST	(HMAC_SESSION_FIRST+MAX_ACTIVE_SESSIONS−1)	最后一个HMAC session
LOADED_SESSION_FIRST	HMAC_SESSION_FIRST	在GetCapability中使用
LOADED_SESSION_LAST	HMAC_SESSION_LAST	在GetCapability中使用
POLICY_SESSION_FIRST	(HR_POLICY_SESSION+0)	第一个policy session
POLICY_SESSION_LAST	(POLICY_SESSION_FIRSTMAX_ACTIVE_SESSIONS−1)	最后一个policy session
TRANSIENT_FIRST	(HR_TRANSIENT+0)	第一个transient对象
ACTIVE_SESSION_FIRST	POLICY_SESSION_FIRST	GetCapability中使用
ACTIVE_SESSION_LAST	POLICY_SESSION_LAST	GetCapability中使用
TRANSIENT_LAST	(TRANSIENT_FIRST+MAX_LOADED_OBJECTS−1)	最后一个transient对象
PERSISTENT_FIRST	(HR_PERSISTENT+0)	第一个persistent对象
PERSISTENT_LAST	(PERSISTENT_FIRST+0x00FFFFFF)	最后一个persistent对象
PLATFORM_PERSISTENT	(PERSISTENT_FIRST+0x00800000)	第一个platform persistent对象
NV_INDEX_FIRST	(HR_NV_INDEX+0)	第一个允许的NV索引
NV_INDEX_LAST	(NV_INDEX_FIRST+0x00FFFFFF)	最后一个NV索引
PERMANENT_FIRST	TCM2_RH_FIRST	
PERMANENT_LAST	TCM2_RH_LAST	

A.3.6 接口类型

A.3.6.1 TCMI_YES_NO

TCMI_YES_NO(BYTE)类型的定义，见表 A.20。

表 A.20 TCMI_YES_NO(BYTE)类型定义

类型	名称
NO	0
YES	1
=TCM2_RC_VALUE	

A.3.6.2 TCMI_DH_OBJECT

TCMI_DH_OBJECT 对象接口类型是引用加载对象的句柄。这里的句柄用于引用临时对象或持久对象,见表 A.21。

表 A.21 TCMI_DH_OBJECT 定义

类型	名称
{TRANSIENT_FIRST:TRANSIENT_LAST}	临时对象的允许范围
{PERSISTENT_FIRST:PERSISTENT_LAST}	持久对象的允许范围
+TCM2_RH_NULL	
=TCM2_RC_VALUE	

A.3.6.3 TCMI_DH_PARENT

TCMI_DH_PARENT 接口类型是一个句柄,该句柄引用可是另一个对象的父对象。此集合中的句柄可引用临时对象或持久对象,也可引用主种子,见表 A.22。

表 A.22 TCMI_DH_PARENT 定义

类型	名称
{TRANSIENT_FIRST:TRANSIENT_LAST}	临时对象的允许范围
{PERSISTENT_FIRST:PERSISTENT_LAST}	持久对象的允许范围
TCM2_RH_OWNER	存储的层次结构
TCM2_RH_PLATFORM	平台的层次结构
TCM2_RH_ENDORSEMENT	背书的层次结构
+TCM2_RH_NULL	—
≠TCM2_RC_VALUE	—

A.3.6.4 TCMI_DH_PERSISTENT

TCMI_DH_PERSISTENT 类型定义,见表 A.23。

表 A.23 TCMI_DH_PERSISTENT 类型定义

值	描述
{PERSISTENT_FIRST:PERSISTENT_LAST}	持续对象允许的范围
=TCM2_RC_VALUE	—

A.3.6.5 TCM2_DH_ENTITY

TCM2_DH_ENTITY 类型定义，见表 A.24。

表 A.24 TCM2_DH_ENTITY 类型定义

值	描述
TCM2_RH_OWNER	—
TCM2_RH_ENDORSEMENT	—
TCM2_RH_PLATFORM	—
TCM2_RH_LOCKOUT	—
{TRANSIENT_FIRST : TRANSIENT_LAST}	对象句柄的范围
{PERSISTENT_FIRST : PERSISTENT_LAST}	—
{NV_INDEX_FIRST : NV_INDEX_LAST}	—
{PCR_FIRST : PCR_LAST}	—
{TCM2_RH_AUTH_00 : TCM2_RH_AUTH_FF}	用户自定义授权值的范围
+TCM2_RH_NULL	条件值
=TCM2_RC_VALUE	—

A.3.6.6 TCMI_DH_PCR

TCMI_DH_PCR 类型定义，见表 A.25。

表 A.25 TCMI_DH_PCR 类型定义

值	描述
{PCR_FIRST:PCR_LAST}	—
+TCM2_RH_NULL	条件值
=TCM2_RC_VALUE	—
{PCR_FIRST:PCR_LAST}	—

A.3.6.7 TCM2_SH_AUTH_SESSION

TCM2_SH_AUTH_SESSION 类型定义，见表 A.26。

表 A.26 TCM2_SH_AUTH_SESSION 类型定义

值	描述
{HMAC_SESSION_FIRST : HMAC_SESSION_LAST}	HMAC授权会话句柄范围
{POLICY_SESSION_FIRST: POLICY_SESSION_LAST}	策略授权会话句柄范围
+TCM2_RS_PW	密码授权
#TCM2_RC_VALUE	句柄超出范围的错误返回值

A.3.6.8 TCM2_SH_HMAC

TCM2_SH_HMAC 类型定义，见表 A.27。

表 A.27 TCM2_SH_HMAC 类型定义

值	描述
{HMAC_SESSION_FIRST: HMAC_SESSION_LAST}	HMAC授权会话句柄范围
=TCM2_RC_VALUE	句柄超出范围的错误返回值

A.3.6.9 TCM2_SH_POLICY

TCM2_SH_POLICY 类型定义，见表 A.28。

表 A.28 TCM2_SH_POLICY 类型定义

值	描述
{POLICY_SESSION_FIRST: POLICY_SESSION_LAST}	策略授权会话句柄范围
=TCM2_RC_VALUE	句柄超出范围的错误返回值

A.3.6.10 TCMI_DH_CONTEXT

TCMI_DH_CONTEXT 类型定义，见表 A.29。

表 A.29 TCMI_DH_CONTEXT 类型定义

值	描述
{HMAC_SESSION_FIRST : HMAC_SESSION_LAST}	HMAC SESSION范围
{POLICY_SESSION_FIRST:POLICY_SESSION_LAST}	POLICY SESSION 范围
{TRANSIENT_FIRST:TRANSIENT_LAST}	TRANSIENT 范围
#TCM2_RC_VALUE	—

A.3.6.11 TCMI_RH_HIERARCHY

TCMI_RH_HIERARCHY 类型定义，见表 A.30。

表 A.30 TCMI_RH_HIERARCH 类型定义

值	描述
TCM2_RH_OWNER	存储等级
TCM2_RH_PLATFORM	平台等级
TCM2_RH_ENDORSEMENT	背书等级
+TCM2_RH_NULL	空
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.12 TCMI_SH_ENABLES

TCMI_SH_ENABLES 类型定义，见表 A.31。

表 A.31 TCMI_SH_ENABLES 类型定义

值	描述
TCM2_RH_OWNER	存储等级
TCM2_RH_PLATFORM	平台等级
TCM2_RH_ENDORSEMENT	背书等级
TCM2_RH_PLATFORM_NV	平台NV
+TCM2_RH_NULL	空
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.13 TCMI_HIERARCHY_AUTH

TCMI_HIERARCHY_AUTH 类型定义，见表 A.32。

表 A.32 TCMI_HIERARCHY_AUTH 类型定义

值	描述
TCM2_RH_OWNER	存储等级
TCM2_RH_PLATFORM	平台等级
TCM2_RH_ENDORSEMENT	背书等级
TCM2_RH_LOCKOUT	锁定授权
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.14 TCMI_RH_PLATFORM

TCMI_RH_PLATFORM 类型定义，见表 A.33。

表 A.33 TCMI_RH_PLATFORM 类型定义

值	描述
TCM2_RH_PLATFORM	平台等级
≠TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.15 TCM2_RH_OWNER

TCM2_RH_OWNER 类型定义，见表 A.34。

表 A.34 TCM2_RH_OWNER 类型定义

值	描述
TCM2_RH_OWNER	所有者等级
＋TCM2_RH_NULL	可为空的句柄
≠TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.16 TCMI_RH_ENDORSEMENT

TCMI_RH_ENDORSEMENT 类型定义，见表 A.35。

表 A.35 TCMI_RH_ENDORSEMENT（TCM2_HANDLE）类型定义

值	描述
TCM2_RH_ENDORSEMENT	背书等级
＋TCM2_RH_NULL	可为空的句柄
≠TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.17 TCMI_RH_PROVISION

TCMI_RH_PROVISION 类型定义，见表 A.36。

表 A.36 TCMI_RH_PROVISION（TCM2_HANDLE）类型定义

值	描述
TCM2_RH_OWNER	所有者授权句柄
TCM2_RH_PLATFORM	平台授权句柄
≠TCM2_RC_VALUE	解析错误时的返回码

A.3.6.18 TCMI_RH_CLEAR

TCMI_RH_CLEAR 类型定义，见表 A.37。

表 A.37 TCMI_RH_CLEAR(TCM2_HANDLE)类型定义

值	描述
TCM2_RH_LOCKOUT	锁定授权的句柄
TCM2_RH_PLATFORM	应用于平台授权的句柄
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.19 TCMI_RH_NV_AUTH

TCMI_RH_NV_AUTH 类型定义，见表 A.38。

表 A.38 TCMI_RH_NV_AUTH(TCM2_HANDLE)类型定义

值	描述
TCM2_RH_PLATFORM	允许平台授权
TCM2_RH_OWNER	允许所有者授权
{NV_INDEX_FIRST:NV_INDEX_LAST}	NV 位置范围
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.20 TCMI_RH_LOCKOUT

TCMI_RH_LOCKOUT 类型定义，见表 A.39。

表 A.39 TCMI_RH_LOCKOUT (TCM2_HANDLE)类型定义

值	描述
TCM2_RH_LOCKOUT	用于锁定授权的句柄
#TCM2_RC_VALUE	解析这个类型错误时候的返回码

A.3.6.21 TCMI_RH_NV_INDEX

TCMI_RH_NV_INDEX 类型定义，见表 A.40。

表 A.40 TCMI_RH_NV_INDEX (TCM2_HANDLE)类型定义

值	描述
{NV_INDEX_FIRST:NV_INDEX_LAST}	NV 索引的范围
#TCM2_RC_VALUE	句柄超出范围时的错误返回值

A.3.6.22 TCMI_ALG_HASH

TCMI_ALG_HASH 类型定义，见表 A.41。

表 A.41 TCMI_ALG_HASH(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.H	杂凑算法
+TCM2_ALG_NULL	—
≠TCM2_RC_HASH	—

A.3.6.23 TCMI_ALG_ASYM(非对称算法)

TCMI_ALG_ASYM 类型定义，见表 A.42。

表 A.42 TCMI_ALG_ASYM(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.AO	所有的非对称对象类型
+TCM2_ALG_NULL	—
≠TCM2_RC_ASYMMETRIC	—

A.3.6.24 TCMI_ALG_SYM(对称算法)

TCMI_ALG_SYM 类型定义，见表 A.43。

表 A.43 TCMI_ALG_SYM(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.S	对称算法密码块
TCM2_ALG_XOR	—
+TCM2_ALG_NULL	—
≠TCM2_RC_SYMMETRIC	—

A.3.6.25 TCMI_ALG_SYM_OBJECT

TCMI_ALG_SYM_OBJECT 类型定义，见表 A.44。

表 A.44 TCMI_ALG_SYM_OBJECT(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.S	对称算法密码块
+TCM2_ALG_NULL	—
≠TCM2_RC_SYMMETRIC	—

A.3.6.26 TCMI_ALG_SYM_MODE

TCMI_ALG_SYM_MODE 类型定义，见表 A.45。

表 A.45 TCMI_ALG_SYM_MODE(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.SE	对称算法密码加密、解密模式
+TCM2_ALG_NULL	—
#TCM2_RC_MODE	—

A.3.6.27 TCMI_ALG_KDF

TCMI_ALG_KDF 类型定义,见表 A.46。

表 A.46 TCMI_ALG_KDF(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_KDF2	
+TCM2_ALG_NULL	
#TCM2_RC_KDF	

A.3.6.28 TCMI_ALG_SIG_SCHEME

TCMI_ALG_SIG_SCHEME 类型定义,见表 A.47。

表 A.47 TCMI_ALG_SIG_SCHEME(TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.ax	所有非对称签名方法,包含匿名方法
TCM2_ALG_HMAC	—
+TCM2_ALG_NULL	—
#TCM2_RC_SCHEME	签名方法不正确的时候的返回值

A.3.6.29 TCMI_ECC_KEY_EXCHANGE

TCMI_ECC_KEY_EXCHANGE 类型定义,见表 A.48。

表 A.48 TCMI_ECC_KEY_EXCHANGE (TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.AM	任何一个ECC密钥的交换方法
TCM2_ALG_SM2	SM2用于签名,但可不可用于密钥交换协议
+TCM2_ALG_NULL	
#TCM2_RC_SCHEME	当密钥交换方法不正确的时候的返回值

A.3.6.30 TCMI_ST_COMMAND_TAG

TCMI_ST_COMMAND_TAG 类型定义，见表 A.49。

表 A.49 TCMI_ST_COMMAND_TAG(TCM2_ST)类型定义

值	描述
TCM2_ST_NO_SESSIONS	非授权标识
TCM2_ST_SESSIONS	授权标识
≠TCM2_RC_BAD_TAG	—
注：结构定义(TCM2_ST)标记用于消除结构的歧义。它们是 16 比特的值。	

A.4 结构定义

A.4.1 TCMS_EMPTY

TCMS_EMPTY 结构用作占位符。

A.4.2 TCMS_ALGORITHM_DESCRIPTION

TCMS_ALGORITHM_DESCRIPTION 结构用于 TCM2_GetCapability()返回预装的算法，见表 A.50。

表 A.50 TCMS_ALGORITHM_DESCRIPTION 结构体定义

参数	类型	描述
alg	TCM2_ALG_ID	密码算法
attributes	TCMA_ALGORITHM	密码算法属性

A.4.3 摘要结构

A.4.3.1 TCMU_HA (Hash)

TCMU_HA 是 TCM 内部计算摘要的杂凑算法联合体，见表 A.51。

表 A.51 TCMU_HA 联合体定义

参数	类型	选择	描述
!ALG.H [!ALG.H_DIGEST_SIZE]	BYTE	TCM2_ALG_!ALG.H	all hashes
null	—	TCM2_ALG_NULL	—

A.4.3.2 TCMT_HA

TCMT_MA 结构是为了处理杂凑灵活性，用 hashAlg 参数来指示用于计算摘要的算法，并指示摘要的大小，见表 A.52。

表 A.52 TCMT_HA 结构体定义

参数	类型	描述
hashAlg	+TCMI_ALG_HASH	摘要选择的算法,包涵了摘要的长度大小
[hashAlg] digest	TCMU_HA	摘要数据

A.4.4 定义的缓存区

A.4.4.1 TCM2B_DIGEST

此结构不能大于 TCM 上实现的杂凑算法生成的最大摘要的缓冲区,见表 A.53。与所有大小的缓冲区一样,检查大小是否在指定范围内。如果不是,则响应代码为 TCM2_RC_SIZE。

表 A.53 TCM2B_DIGEST 结构体定义

参数	类型	描述
size	UINT16	摘要缓存区大小,可为 0
buffer[size]{:sizeof(TCMU_HA)}	BYTE	保持摘要的缓存区

A.4.4.2 TCM2B_DATA

此结构用于对象名称大小的数据缓冲区,见表 A.54。

表 A.54 TCM2B_DATA 结构体定义

参数	类型	描述
size	UINT16	摘数据缓存区大小,可为 0
buffer[size]{:sizeof(TCMT_HA)}	BYTE	—

A.4.4.3 TCM2B_NONCE

TCM2B_NONCE 结构体定义见表 A.55。

表 A.55 TCM2B_NONCE 结构体定义

参数	类型	描述
TCM2B_DIGEST	TCM2B_NONCE	限制为与摘要结构相同

A.4.4.4 TCM2B_AUTH

此结构用于授权值,并将 authValue 设置为 TCM 生成的最大摘要,见表 A.56。为了确保对象内的一致性,authValue 不能大于对象 nameAlg 生成的摘要的大小。

表 A.56 TCM2B_AUTH 结构体定义

参数	类型	描述
TCM2B_DIGEST	TCM2B_AUTH	限制为与摘要结构相同

A.4.4.5 TCM2B_OPERAND

此结构保存操作数以便与 NV 索引位置进行比较，见表 A.57。

表 A.57 TCM2B_OPERAND 结构体定义

参数	类型	描述
TCM2B_DIGEST	TCM2B_OPERAND	限制为与摘要结构相同

A.4.4.6 TCM2B_EVENT

此结构是事件数据的大小缓冲区，见表 A.58。

表 A.58 TCM2B_EVENT 结构体定义

参数	类型	描述
size	UINT16	事件数据大小
buffer [size] {:1024}	BYTE	事件数据

A.4.4.7 TCM2B_MAX_BUFFER

此结构可为使用大数据缓冲区[如 TCM2_Hash()、TCM2_SequenceUpdate()]的命令保留最大的缓冲区，见表 A.59。

表 A.59 TCM2B_MAX_BUFFER 结构体定义

参数	类型	描述
size	UINT16	数据大小
buffer [size] {:MAX_DIGEST_BUFFER}	BYTE	数据

A.4.4.8 TCM2B_MAX_NV_BUFFER

此结构应用于 NV 的最大缓冲区，见表 A.60。

表 A.60 TCM2B_MAX_NV_BUFFER 结构体定义

参数	类型	描述
size	UINT16	Buffer的大小
buffer [size] {:MAX_NV_BUFFER_SIZE}	BYTE	操作数 MAX_NV_BUFFER_SIZE 依赖 TCM

A.4.4.9 TCM2B_TIMEOUT

此结构用于提供授权的超时值，见表 A.61。

表 A.61 TCM2B_TIMEOUT 结构体定义

参数	类型	描述
size	UINT16	超时值大小
buffer[size]{:sizeof(UINT64)}	BYTE	超时值

A.4.4.10 TCM2B_IV

此结构用于传递对称分组密码的 IV 初始值，见表 A.62。

表 A.62 TCM2B_IV 结构体定义

参数	类型	描述
size	UINT16	IV 值大小，固定值
buffer[size]{:MAX_SYM_BLOCK_SIZE}	BYTE	IV 值

A.4.5 实体名称

A.4.5.1 TCMU_NAME

TCMU_NAME 结构定义见表 A.63。

表 A.63 TCMU_NAME 结构体定义

参数	类型	描述
digest	TCMT_HA	当名称是摘要
handle	TCM2_HANDLE	当名称是句柄

A.4.5.2 TCM2B_NAME

此缓冲区保存任何实体类型的名称，见表 A.64。

结构中的名称类型由上下文和大小参数决定。如果大小为 4，则名称为句柄。如果大小为 0，则不存在名称。

表 A.64 TCM2B_NAME 结构体定义

参数	类型	描述
size	UINT16	名称结构大小
name[size]{:sizeof(TCMU_NAME)}	BYTE	名称结构数据

A.4.6 PCR结构

A.4.6.1 TCMS_PCR_SELECT

该结构提供了一种标准方法来指定 PCR 列表，见表 A.65。PCR 编号从 0 开始。

pcrSelect 是一个八比特字节的数组。通过将 PCR 数除以 8，找到包含与特定 PCR 对应的比特的八比特字节。

最小：

$$PCR\_SELECT\_MIN = (PLATFORM\_PCR+7)/8$$

PLATFORM_PCR：平台特定规范要求的 PCR 数量，最小为 24。

最大：

$$PCR\_SELECT\_MAX = (IMPLEMENTATION\_PCR+7)/8$$

IMPLEMENTATION_PCR：TCM 声明的最大的 PCR 数，由厂商自行定义。

表 A.65 TCMS_PCR_SELECT 结构体定义

参数	类型	描述
sizeofSelect {PCR_SELECT_MIN:}	UINT8	PCR数组大小
pcrSelect [sizeofSelect] {:PCR_SELECT_MAX}	BYTE	所选PCR的数据
≠TCM2_RC_VALUE	—	—

A.4.6.2 TCMS_PCR_SELECTION

TCMS_PCR_SELECTION 结构体定义，见表 A.66。

表 A.66 TCMS_PCR_SELECTION 结构体定义

参数	类型	描述
hash	TCMI_ALG_HASH	选择关联的杂凑算法
sizeofSelect {PCR_SELECT_MIN:}	UINT8	pcrSelect数组的大小
pcrSelect [sizeofSelect] {:PCR_SELECT_MAX}	BYTE	所选PCR的数据
≠TCM2_RC_VALUE	—	—

A.4.7 Tickets结构

A.4.7.1 概述

票证是 TCM 以前处理过一些信息的证据。票证是使用只有 TCM 知道的密钥对数据进行 HMAC。票证是一种扩展 TCM 状态内存的方法。票证仅由生成它的 TCM 使用。

符合本规范的 TCM 将使用本条中所示的票据配方，见表 A.67。

创建票证数据的方法是：

创建票证数据的方法是：

$$HMAC_{contexAlg}(proof, (ticketType \| param\{\| param\{\cdots\}\}))$$

式中：

$HMAC_{contexAlg}()$ ——一个使用杂凑 HMAC,用于上下文完整性;
proof ——TCM 的秘密值(取决于层次结构);
ticketType ——区分票证的值;
param ——由 TCM 检查的一个或者多个值。

表 A.67 Tickets 中 proof 的值

层次	证明	描述
Null	nullProof	随每次TCM重置而更改的值
Platform	phProof	随PPS的每次变化而变化的值
Owner	shProof	随SPS的每次变化而变化的值
Endorsement	ehProof	随EPS或SPS的每次变化而变化的值

A.4.7.2 TCMT_TK_CREATION

此票证由 TCM2_Create()或 TCM2_CreatePrimary()生成,见表 A.68。它用于将创建数据绑定到应用它的对象。

表 A.68 TCMS_TK_CREATION 结构体定义

参数	类型	描述
tag {TCM2_ST_CREATION}	TCM2_ST	票据标签
=TCM2_RC_TAG	—	标记未创建TCM存储时返回错误
hierarchy	TCMI_RH_HIERARCHY+	包含名称的层次结构
digest	TCM2B_DIGEST	使用层次证明值生成的HMAC值

A.4.7.3 TCMT_TK_VERIFIED

此票证由 TCM2_VerifySignature()的函数生成,见表 A.69。此公式用于多种票据用途。

表 A.69 TCMS_TK_VERIFIED 结构体定义

参数	类型	描述
tag {TCM2_ST_VERIFIED}	TCM2_ST	票据标签
=TCM2_RC_TAG	—	未验证标记时返回错误
hierarchy	TCMI_RH_HIERARCHY+	密钥名称层次结构
digest	TCM2B_DIGEST	使用层次证明值生成的HMAC值

A.4.7.4 TCMT_TK_AUTH

当授权有过期时间时,此票证由 TCM2_PolicySigned()和 TCM2_PolicySecret()生成,见表 A.70。

表 A.70 TCMS_TK_AUTH 结构体定义

参数	类型	描述
tag {TCM2_ST_AUTH_SIGNED, TCM2_ST_AUTH_SECRET}	TCM2_ST	票据标签
#TCM2_RC_TAG		未验证标记时返回错误
hierarchy	TCMI_RH_HIERARCHY+	生成票证的对象的层次结构
digest	TCM2B_DIGEST	使用层次证明值生成的HMAC值

A.4.7.5 TCMT_TK_HASHCHECK

TCMS_TK_HASHCHECK 结构体定义，见表 A.71。

表 A.71 TCMS_TK_HASHCHECK 结构体定义

参数	类型	描述
tag {TCM2_ST_HASHCHECK}	TCM2_ST	票据标签
#TCM2_RC_TAG	—	不是TCM ST_HASHCHECK时返回错误
hierarchy	TCMI_RH_HIERARCHY+	层次结构
digest	TCM2B_DIGEST	使用层次证明值生成的HMAC值

A.4.8 属性

A.4.8.1 TCMS_ALG_PROPERTY

此结构用于报告算法标识符的属性，见表 A.72。

表 A.72 TCMS_ALG_PROPERTY 结构体定义

参数	类型	描述
alg	TCM2_ALG_ID	算法标识符
algProperties	TCMA_ALGORITHM	算法的属性

A.4.8.2 TCMS_TAGGED_PROPERTY

此结构用于报告 UINT32 值的属性，见表 A.73，为 TCM2_GetCapability()返回的值。

表 A.73 TCMS_TAGGED_PROPERTY 结构体定义

参数	类型	描述
property	TCM2_PT	属性标识符
value	UINT32	属性值

A.4.8.3 TCMS_TAGGED_PCR_SELECT

此结构在 TCM2_GetCapability()中用于返回 PCR 的属性，见表 A.74。

表 A.74 TCMS_TAGGED_PCR_SELECT 结构体定义

参数	类型	描述
tag	TCM2_PT_PCR	属性标识符
sizeofSelect {PCR_SELECT_MIN:}	UINT8	pcrSelect数组的大小
pcrSelect [sizeofSelect] {:PCR_SELECT_MAX}	BYTE	具有识别特性的PCR值

A.4.8.4 TCMS_TAGGED_POLICY

此结构在 TCM2_GetCapability()中用于返回与永久句柄关联的策略，见表 A.75。

表 A.75 TCMS_TAGGED_POLICY 结构体定义

参数	类型	描述
handle	TCM2_HANDLE	永久句柄
policyHash	TCMT_HA	策略算法与杂凑

A.4.8.5 TCMA_MODES

此结构定义了 TCM 应符合设计的模式，见表 A.76。该结构通过 TCM2_GetCapability(capability = TCM2_CAP_TCM2_PROPERTIES, property = TCM2_PT_MODES)读取内容。

表 A.76 TCMA_MODES 结构定义

比特	名称	定义
15:0	保留	0
24:16	CCTC	TCM实例体现其检测认证的结果
31:25	保留	0

A.4.9 列表

A.4.9.1 TCML_CC

TCML_CC 结构体定义，见表 A.77。

表 A.77 TCML_CC 结构体定义

参数	类型	描述
count	UINT32	在命令列表里面命令的个数，可为0
commandCodes[count]{:MAX_CAP_CC}	TCM2_CC	命令码列表 最大值仅应用于在一个命令的命令列表里，返回的大小被限定于buffer参数的大小
≠TCM2_RC_SIZE	—	当个数大于列表允许的最大值得时候的返回值

A.4.9.2 TCML_CCA

TCML_CCA 结构体定义，见表 A.78。

表 A.78 TCML_CCA 结构体定义

参数	类型	描述
count	UINT32	命令属性列表的值的大小，可为0
commandAttributes[count]{:MAX_CAP_CC}	TCMA_CC	命令码属性列表

A.4.9.3 TCML_ALG

TCML_ALG 结构体定义，见表 A.79。

表 A.79 TCML_ALG 结构体定义

参数	类型	描述
count	UINT32	在算法列表里面的算法个数，可为0
algorithms[count]{:MAX_ALG_LIST_SIZE}	TCM2_ALG_ID	算法ID列表 最大值仅应用于在一个命令的命令列表里，返回的大小被限定于buffer参数的大小
≠TCM2_RC_SIZE	—	当个数大于列表允许的最大值得时候的返回值

A.4.9.4 TCML_HANDLE

TCML_HANDLE 结构体定义，见表 A.80。

表 A.80 TCML_HANDLE 结构体定义

参数	类型	描述
count	UINT32	列表里句柄的个数 可为0
handle[count]{: MAX_CAP_HANDLES}	TCM2_HANDLE	句柄数组
≠TCM2_RC_SIZE	—	当个数大于列表允许的最大值得时候的返回值

A.4.9.5 TCML_DIGEST

TCML_DIGEST 结构体定义，见表 A.81。

表 A.81 TCML_DIGEST 结构体定义

参数	类型	描述
count {2:}	UINT32	列表里摘要的个数，对于TCM2_PolicyOR()最小为2
digests[count]{:8}	TCM2B_DIGEST	摘要列表
≠TCM2_RC_SIZE	—	当个数大于列表允许的最大值得时候的返回值

A.4.9.6 TCML_DIGEST_VALUES

TCML_DIGEST_VALUES 结构体定义，见表 A.82。

表 A.82 TCML_DIGEST_VALUES 结构体定义

参数	类型	描述
count	UINT32	列表里的摘要个数
digests[count]{:HASH_COUNT}	TCMT_HA	被标识摘要的列表
≠TCM2_RC_SIZE	—	当个数大于可能的bank的个数的返回值

A.4.9.7 TCM2B_DIGEST_VALUES

TCM2B_DIGEST_VALUES 结构体定义，见表 A.83。

表 A.83 TCM2B_DIGEST_VALUES 结构体定义

参数	类型	描述
size	UINT16	运算的大小
buffer [size] {:sizeof(TCML_DIGEST_VALUES)}	BYTE	运算

A.4.9.8 TCML_PCR_SELECTION

TCML_PCR_SELCETION 结构体定义，见表 A.84。

表 A.84 TCML_PCR_SELECTION 结构体定义

参数	类型	描述
count	UINT32	选择的结构体个数 可为0
pcrSelections[count]{:HASH_COUNT}	TCMS_PCR_SELECTION	选择的列表
≠TCM2_RC_SIZE		当个数大于可能的bank的个数的返回值

A.4.9.9 TCML_ALG_PROPERTY

TCML_ALG_PROPERTY 结构体定义，见表 A.85。

表 A.85 TCML_ALG_PROPERTY 结构体定义

参数	类型	描述
count	UINT32	算法属性结构体个数 可为0
algProperties[count]{:MAX_CAP_ALGS}	TCMS_ALG_PROPERTY	属性列表

A.4.9.10 TCML_TAGGED_TCM2_PROPERTY

TCML_TAGGED_TCM2_PROPERTY 结构体定义，见表 A.86。

表 A.86 TCML_TAGGED_TCM2_PROPERTY 结构体定义

参数	类型	描述
count	UINT32	属性个数 可为0
TCMProperty[count]{:MAX_TCM2_PROPERTIES}	TCMS_TAGGED_PROPERTY	被标识属性的数组

A.4.9.11 TCML_TAGGED_PCR_PROPERTY

TCML_TAGGED_PCR_PROPERTY 结构体定义，见表 A.87。

表 A.87 TCML_TAGGED_PCR_PROPERTY 结构体定义

参数	类型	描述
count	UINT32	属性个数 属性为0
pcrProperty[count]{:MAX_PCR_PROPERTIES}	TCMS_TAGGED_PCR_SELECT	被标识的PCR选择

A.4.9.12 TCML_ECC_CURVE

TCML_ECC_CURVE 结构体定义，见表 A.88。

表 A.88 TCML_ECC_CURVE 结构体定义

参数	类型	描述
count	UINT32	曲线个数可为0
eccCurves[count]{:MAX_ECC_CURVES}	TCM2_ECC_CURVE	确定的ECC曲线数组

A.4.10 CAPABILITIES 结构

A.4.10.1 TCMU_CAPABILITIES

TCMU_CAPABILITIES 联合体定义，见表 A.89。

表 A.89 TCMU_CAPABILITIES 联合体定义

参数	类型	选择器	描述
algorithms	TCML_ALG_PROPERTY	TCM2_CAP_ALGS	—
handles	TCML_HANDLE	TCM2_CAP_HANDLES	—
command	TCML_CCA	TCM2_CAP_COMMANDS	—
ppCommands	TCML_CC	TCM2_CAP_PP_COMMANDS	—

表 A.89 TCMU_CAPABILITIES 联合体定义（续）

参数	类型	选择器	描述
auditCommands	TCML_CC	TCM2_CAP_AUDIT_COMMANDS	—
assignedPCR	TCML_PCR_SELECTION	TCM2_CAP_PCRS	—
tcmProperties	TCML_TAGGED_TCM2_PROPERTY	TCM2_CAP_TCM2_PROPERTIES	—
pcrProperties	TCML_TAGGED_PCR_PROPERTY	TCM2_CAP_PCR_PROPERTIES	—
eccCurves	TCML_ECC_CURVE	TCM2_CAP_ECC_CURVES	TCM2_ALG_ECC

A.4.10.2 TCMS_CAPABILITY_DATA

TCMS_CAPABILITY_DATA 结构体定义，见表 A.90。

表 A.90 TCMS_CAPABILITY_DATA 结构体定义

参数	类型	描述
capability	TCM2_CAP	—
[capability]data	TCMU_CAPABILITIES	capability 数据

A.4.11 Clock/Counter 结构

A.4.11.1 TCMS_CLOCK_INFO

TCMS_CLOCK_INFO 结构体定义，见表 A.91。

表 A.91 TCMS_CLOCK_INFO 结构体定义

参数	类型	描述
clock	UINT64	TCM 通电时前进的时间值(毫秒)
resetCount	UINT32	自上次 TCM2_Clear()以来 TCM 重置的次数
restartCount	UINT32	自上次重置或 TCM2_Clear()之后，TCM2_Shutdown()已发生的次数
safe	TCMI_YES_NO	TCM 以前没有报告过大于当前时钟值的时钟值

A.4.11.2 TCMS_TIME_INFO

该结构用于 TCM2_GetTime()，见表 A.92。每当重新建立时间电路的电源时，此结构中报告的时间值将被重置。根据实际情况，实现可在 TCM2_Startup()之后 TCM 返回之前的任何时间重置时间值。当 TCM 通电时，时间值应持续增加。

表 A.92 TCMS_TIME_INFO 结构体定义

参数	类型	描述
time	UINT64	电路上次复位后的时间(毫秒)
clockInfo	TCMS_CLOCK_INFO	包含时钟信息的结构

A.4.12 证明结构

A.4.12.1 TCMS_TIME_ATTEST_INFO

当执行 TCM2_GetTime()时使用此结构,见表 A.93。

表 A.93 TCMS_TIME_ATTEST_INFO 结构体定义

参数	类型	描述
time	TCMS_TIME_INFO	时间、时钟、重置计数、重新启动计数和安全指示器
firmwareVersion	UINT64	TCM供应商设置的固件版本号

A.4.12.2 TCMS_CERTIFY_INFO

这是 TCM2_Certify()的证明数据,见表 A.94。

表 A.94 TCMS_CERTIFY_INFO 结构体定义

参数	类型	描述
name	TCM2B_NAME	证明对象名称
qualifiedName	TCM2B_NAME	证明对象的限定名称

A.4.12.3 TCMS_QUOTE_INFO

TCM2_Quote()的证明数据结构,见表 A.95。

表 A.95 TCMS_QUOTE_INFO 结构体定义

参数	类型	描述
pcrSelect	TCML_PCR_SELECTION	算法、PCR和摘要等数据信息
pcrDigest	TCM2B_DIGEST	所选PCR的摘要签名密钥的杂凑计算

A.4.12.4 TCMS_CREATION_INFO

TCM2_CertifyCreation()的证明数据结构,见表 A.96。

表 A.96 TCMS_CREATION_INFO 结构体定义

参数	类型	描述
objectName	TCM2B_NAME	对象名称
creationHash	TCM2B_DIGEST	Creation的摘要值

A.4.12.5 TCMI_ST_ATTEST

TCMI_ST_ATTEST 类型定义,见表 A.97。

表 A.97 TCMI_ST_ATTEST 类型定义

值	描述
TCM2_ST_ATTEST_CERTIFY	TCM2_Certify()产生
TCM2_ST_ATTEST_QUOTE	TCM2_Quote()产生
TCM2_ST_ATTEST_TIME	TCM2_GetTime()产生
TCM2_ST_ATTEST_CREATION	TCM2_CertifyCreation()产生

A.4.12.6 TCMU_ATTEST

TCMU_ATTEST 联合体定义见表 A.98。

表 A.98 TCMU_ATTEST 联合体定义

参数	类型	值
certify	TCMS_CERTIFY_INFO	TCM2_ST_ATTEST_CERTIFY
creation	TCMS_CREATION_INFO	TCM2_ST_ATTEST_CREATION
quote	TCMS_QUOTE_INFO	TCM2_ST_ATTEST_QUOTE
time	TCMS_TIME_ATTEST_INFO	TCM2_ST_ATTEST_TIME
nv	TCMS_NV_CERTIFY_INFO	TCM2_ST_ATTEST_NV_DIGEST

A.4.12.7 TCMS_ATTEST

TCMS_ATTEST 结构体定义,见表 A.99。

表 A.99 TCMS_ATTEST 结构体定义

参数	类型	描述
magic	TCM2_GENERATED	表示由TCM创建的结构
type	TCMI_ST_ATTEST	证明结构类型
qualifiedSigner	TCM2B_NAME	签名密钥的名称
extraData	TCM2B_DATA	调用者提供的外部信息
clockInfo	TCMS_CLOCK_INFO	时钟、重置计数、重新启动计数和安全
firmwareVersion	UINT64	TCM供应商提供的固件版本号
[type]attested	TCMU_ATTEST	特定类型的证明信息

A.4.12.8 TCM2B_ATTEST

此缓冲区包含签名结构，见表 A.100。证明数据是结构的签名部分。

表 A.100 TCM2B_ATTEST 结构体定义

参数	类型	描述
size	UINT16	证明数据结构的大小
attestationData[size]{:sizeof(TCMS_ATTEST)}	BYTE	签名数据结构

A.4.13 授权结构

A.4.13.1 TCMS_AUTH_COMMAND

该结构体用于命令会话区域中每个授权的结构，见表 A.101。

表 A.101 TCMS_AUTH_COMMAND 结构体定义

参数	类型	描述
sessionHandle	TCMI_SH_AUTH_SESSION+	会话句柄
nonce	TCM2B_NONCE	会话 nonce，可以是空缓冲区
sessionAttributes	TCMA_SESSION	会话属性
hmac	TCM2B_AUTH	HMAC，密码或空授权

A.4.13.2 TCMS_AUTH_RESPONSE

该结构体用于响应会话区域中每个授权的结构，见表 A.102。如果 TCM 返回 TCM2_RC_SUCCESS，则响应的会话区域包含与命令相同数量的授权，并且授权的顺序相同。

表 A.102 TCMS_AUTH_RESPONSE 结构体定义

参数	类型	描述
nonce	TCM2B_NONCE	会话 nonce，可以是空缓冲区
sessionAttributes	TCMA_SESSION	会话属性
hmac	TCM2B_AUTH	一个 HMAC 或一个空授权

A.5 密码参数和结构

A.5.1 对称算法

A.5.1.1 TCMI_!ALG.S_KEY_BITS

TCMI_!ALG.S_KEY_BITS 类型定义，见表 A.103，密钥长度以比特(bit)表示。

表 A.103 TCMI_!ALG.S_KEY_BITS 类型定义

参数	描述
$!ALG.S_KEY_SIZES_BITS	密钥长度(单位:bit)
≠TCM2_RC_VALUE	当密钥长度不支持的时候返回的错误

A.5.1.2 TCMU_SYM_KEY_BITS

TCMU_SYSM_KEY_BITS 联合体的定义,见表 A.104。

表 A.104 TCMU_SYSM_KEY_BITS 联合体定义

参数	类型	选择器	描述
!ALG.S	TCMI_!ALG.S_KEY_BITS	TCM2_ALG_!ALG.S	对称算法
sym	TCM2_KEY_BITS		当选择器可以是任何对称分组密码时
xor	TCMI_ALG_HASH	TCM2_ALG_XOR	使用异或的重载 注意:不允许 TCM2_ALG_NULL
null	—	TCM2_ALG_NULL	—

A.5.1.3 TCMU_SYM_MODE

TCMU_SYM_MODE 联合体的定义,见表 A.105。

表 A.105 TCMU_SYM_MODE 联合体定义

参数	类型	选择器	描述
!ALG.S	TCMI_ALG_SYM_MODE	TCM2_ALG_!ALG.S	—
sym	TCMI_ALG_SYM_MODE		当选择器可以是任何对称分组密码时
xor	—	TCM2_ALG_XOR	无模式选择器
null	—	TCM2_ALG_NULL	无模式选择器

A.5.1.4 TCMT_SYM_DEF_OBJECT

TCMT_SYM_DEF_OBJECT 结构体定义,见表 A.106。

表 A.106 TCMT_SYM_DEF_OBJECT 结构体定义

参数	类型	描述
algorithm	+TCMI_ALG_SYM_OBJECT	选择对称分组密码
[algorithm]keyBits	TCMU_SYM_KEY_BITS	密钥大小
[algorithm]mode	TCMU_SYM_MODE	默认模式
//[algorithm]details	TCMU_SYM_DETAILS	包含其他算法详细信息(如果有)

A.5.1.5 TCMU_SYM_DETAILS

TCMU_SYM_DETAILS 联合体的定义，见表 A.107。

表 A.107 TCMU_SYM_DETAILS 联合体定义

参数	类型	选择器	描述
!ALG.S	—	TCM2_ALG_!ALG	—
sym	—	—	当选择器可以是任何对称分组密码时
xor	—	TCM2_ALG_XOR	—
null	—	TCM2_ALG_NULL	—

A.5.1.6 TCM2B_SYM_KEY

此结构用于在非对称对象的敏感区域中保存对称密钥，见表 A.108。

密钥中的比特数以公共区域中的密钥比特数为单位。当密钥比特数不是 8 位的偶数倍时，缓冲区的未使用位将是缓冲区的有效位。

表 A.108 TCM2B_SYM_KEY 结构体定义

参数	类型	描述
size	UINT16	包含密钥的缓冲区的大小(八比特字节)；可为 0
buffer [size] {:MAX_SYM_KEY_BYTES}	BYTE	密钥

A.5.1.7 TCM2B_DERIVE

TCM2B_DERIVE 结构体定义，见表 A.109。

表 A.109 TCM2B_DERIVE 结构体定义

参数	类型	描述
size	UINT16	缓冲区的大小
buffer[size]{: sizeof(TCMS_DERIVE)}	BYTE	已创建对象的对称数据或派生对象的标签和上下文

A.5.1.8 TCM2B_SENSITIVE_CREATE

此结构在大小合适的缓冲区中包含敏感的创建数据，见表 A.110。

定义此结构是为了使 TCMS_SENSITIVE_CREATE 的 userAuth 和数据值都可作为参数加密的单个参数传递。

表 A.110 TCM2B_SENSITIVE_CREATE 结构体定义

参数	类型	描述
size=	UINT16	敏感字节的大小(不能为0) “=”表示TCM来验证结构体的其余部分的长度,也就是缓冲区中用于成功解组结构体的字节数应与size相同
sensitive	TCMS_SENSITIVE_CREATE	要密封的数据或对称密钥值

A.5.1.9 TCM2B_SENSITIVE_DATA

TCM2B_SENSITIVE_DATA 结构体定义,见表 A.111。

缓冲区保存数据对象的秘密数据。它可以保存多达 128 个八比特字节的数据,MAX_SYM_DATA 值应为 128。

表 A.111 TCM2B_SENSITIVE_DATA 结构体定义

参数	类型	描述
size	UINT16	—
buffer[size]{: MAX_SYM_DATA}	BYTE	键控杂凑私有数据结构

A.5.2 非对称算法

A.5.2.1 TCMU_KDF_SCHEME

TCMU_KDF_SCHEME 的定义,见表 A.112。

表 A.112 TCMU_KDF_SCHEME 联合体定义

参数	类型	选择器	描述
kdf2	TCMS_SCHEME_KDF2	TCM2_ALG_KDF2	
null		TCM2_ALG_NULL	

A.5.2.2 TCMT_KDF_SCHEME

TCMT_KDF_SCHEME 的定义,见表 A.113。

表 A.113 TCMT_KDF_SCHEME 结构体定义

参数	类型	描述
scheme	+TCMI_ALG_KDF	scheme选择器
[scheme]details	TCMU_KDF_SCHEME	scheme参数

A.5.2.3 TCM2_ECC_PARAMETER

TCM2B_ECC_PARAMETER 结构体定义见表 A.114。此大小的缓冲区包含 TCM 支持的最大 ECC 参数(坐标)。

表 A.114 TCM2B_ECC_PARAMETER 结构体定义

参数	类型	描述
size	UINT16	buffer的大小
buffer[size]{:MAX_ECC_KEY_BYTES}	BYTE	数据

A.5.2.4 TCMS_ECC_POINT

TCMS_ECC_POINT 结构体定义,见表 A.115。

这个结构包含两个 ECC 坐标,它们一起构成一个 ECC 点。

表 A.115 TCMS_ECC_POINT 结构体定义

参数	类型	描述
x	TCM2B_ECC_PARAMETER	*X* 坐标
y	TCM2B_ECC_PARAMETER	*Y* 坐标

A.5.2.5 TCM2B_ECC_POINT

此大小的缓冲区包含 TCM 支持的最大 ECC 参数(坐标),见表 A.116。

表 A.116 TCM2B_ECC_POINT 结构体定义

参数	类型	描述
size	UINT16	数据区大小
buffer[size]{:MAX_ECC_KEY_BYTES}	BYTE	参数数据

A.5.3 签名

A.5.3.1 TCMU_SIGNATURE

TCMU_SIGNATURE_COMPOSITE 是特定 TCM 实现所支持的各种签名的联合,见表 A.117。

表 A.117 TCMU_SIGNATURE 结构体定义

参数	类型	选择值	描述
!ALG.ax	TCMS_SIGNATURE_!ALG.ax	TCM_ALG_!ALG.ax	所有非对称签名
hmac	TCMT_HA	TCM2_ALG_HMAC	HMAC签名
any	TCMS_SCHEME_HASH		用于访问杂凑算法
null	—	TCM2_ALG_NULL	空签名

A.5.3.2 TCMT_SIGNATURE

显示了表示对称或非对称签名时的基本算法灵活结构，定义见表 A.118。sigAlg 参数表示用于签名的算法。

表 A.118 TCMT_SIGNATURE 结构体定义

参数	类型	描述
sigAlg	+TCMI_ALG_SIG_SCHEME	用于构造签名的算法的选择算法
[sigAlg]signature	TCMU_SIGNATURE	应为实际的签名信息

A.5.4 秘密交换

A.5.4.1 TCMU_ENCRYPTED_SECRET

TCMU_ENCRYPTED_SECRET 定义，见表 A.119。

表 A.119 TCMU_ENCRYPTED_SECRET 结构体定义

参数	类型	选择值	描述
ecc[sizeof(TCMS_ECC_POINT)]	BYTE	TCM2_ALG_ECC	—
symmetric[sizeof(TCM2B_DIGEST)]	BYTE	TCM2_ALG_SYMCIPHER	—
keyedHash[sizeof(TCM2B_DIGEST)]	BYTE	TCM2_ALG_KEYEDHASH	任何对称加密的秘密值将被限制为不大于摘要

A.5.4.2 TCM2B_ENCRYPTED_SECRET

TCM2B_ENCRYPTED_SECRET 定义，见表 A.120。

表 A.120 TCM2B_ENCRYPTED_SECRET 结构体定义

参数	类型	描述
size	UNIT16	秘密值的大小
secret[size]{:sizeof(TCMU_ENCRYPTED_SECRET)}	BYTE	秘密

A.6 密钥/对象结构

A.6.1 公钥数据结构

A.6.1.1 TCMI_ALG_PUBLIC

TCMI_ALG_PUBLIC 定义，见表 A.121。

表 A.121 TCMI_ALG_PUBLIC 结构体定义

值	描述
TCM_ALG_!ALG.o	所有对象类型
=TCM2_RC_TYPE	不支持公共类型时的响应代码

A.6.1.2 TCMU_PUBLIC_ID

TCMT_PUBLIC 的字段中允许的所有值的并集，见表 A.122。

表 A.122 TCMU_PUBLIC_ID 结构体定义

参数	类型	值	描述
keyedHash	TCM2B_DIGEST	TCM2_ALG_KEYEDHASH	—
sym	TCM2B_DIGEST	TCM2_ALG_SYMCIPHER	—
ecc	TCMS_ECC_POINT	TCM2_ALG_ECC	—
derive	TCMS_DERIVE	—	仅当parentHandle是派生父级时，才允许TCM2_CreateLoaded

A.6.1.3 TCMS_ASYM_PARMS

此结构包含非对称密钥的公共区域参数，见表 A.123。

表 A.123 TCMS_ASYM_PARMS 结构体定义

参数	类型	描述
symmetric	TCMT_SYM_DEF_OBJECT+	限制解密密钥的伴随对称算法，并应设置为支持的对称算法
scheme	TCMT_ASYM_SCHEME+	对于具有签名属性的密钥，该密钥类型的有效签名方案。 对于具有加密属性的密钥，有效的密钥交换协议

A.6.1.4 TCMS_ECC_PARMS

该结构包含 SM2 和 ECC 非对称密码算法的参数，见表 A.124。

表 A.124 TCMS_ECC_PARMS 结构体定义

参数	类型	描述
symmetric	TCMT_SYM_DEF_OBJECT+	对于用途受限解密密钥，应设置为支持的对称算法，密钥大小和模式。 如果密钥不是受限解密密钥，则此字段应设置为 TCM2_ALG_NULL
scheme	TCMT_ECC_SCHEME+	如果设置了密钥的签名属性，则这将是一个有效的签名方案。 如果设置了密钥的加密属性，则这应是有效的密钥交换方案或 TCM2_ALG_NULL。 如果密钥是存储密钥，则此字段应为 TCM2_ALG_NULL
curveID	TCMI_ECC_CURVE	非对称算法曲线 ID
kdf	TCMT_KDF_SCHEME+	从Z值生成对称密钥的可选密钥派生方案。 如果与curveID关联的kdf参数不是TCM2_ALG_NULL，则此参数应为NULL

A.6.1.5 TCMU_PUBLIC_PARMS

定义了可能包含在密钥的公共部分中的参数定义结构，见表 A.125。

表 A.125 TCMU_PUBLIC_PARMS 结构体定义

参数	类型	选择符	描述
keyedHashDetail	TCMS_KEYEDHASH_PARMS	TCM2_ALG_KEYEDHASH	签名|解密|全不(2)
symDetail	TCMS_SYMCIPHER_PARMS	TCM2_ALG_SYMCIPHER	签名|解密|全不(2)
eccDetail	TCMS_ECC_PARMS	TCM2_ALG_ECC	解密＋ 签名(2)
asymDetail	TCMS_ASYM_PARMS	—	ECC密钥的通用方案结构
注1：描述的列指明TCMA_OBJECT.decrypt或TCMA_OBJECT.sign哪个可以设置。 注2："＋"表示两个都可以设置，但应设置一个，"|"表示可选设置。			

A.6.1.6 TCMT_PUBLIC

定义公共区域结构。使用算法名称将其与此结构的摘要连接起来，见表 A.126。

表 A.126 TCMT_PUBLIC 结构体定义

参数	类型	描述
type	TCMI_ALG_PUBLIC	与此对象关联的"算法"
nameAlg	＋TCMI_ALG_HASH	用于计算对象名称的算法
objectAttributes	TCMA_OBJECT	连同类型一起的属性信息，决定这个对象的操作
authPolicy	TCM2B_DIGEST	使用此密钥的可选策略
[type]parameters	TCMU_PUBLIC_PARMS	算法或结构细节
[type]unique	TCMU_PUBLIC_ID	结构的唯一标识符。对于非对称密钥，将是公钥

A.6.1.7 TCM2B_PUBLIC

此大小的缓冲区用于在加载命令和返回公共区域的任何响应中嵌入 TCMT_PUBLIC，见表 A.127。

表 A.127 TCM2B_PUBLIC 结构体定义

参数	类型	描述
size=	UINT16	publicArea大小 如果TCMT_PUBLIC的所有必需字段都不存在，则在尝试解开TCMT_PUBLIC时，TCM将返回一个错误(通常为TCM2_RC_SIZE) "="表示由TCM来验证结构体的其余部分的长度，也就是缓冲区中用于成功解组结构体的字节数应与size相同
publicArea	＋TCMT_PUBLIC	公共结构数据 注："＋"表示调用方可指定允许对nameAlg使用TCM2_ALG_NULL

A.6.1.8 TCM2B_TEMPLATE

此大小的缓冲区用于为 TCM2_CreateLoaded()嵌入 TCMT_TEMPLETE 模板，见表 A.128。

由于对向后兼容性的要求，此结构的解组相当复杂。与 TCM2B_PUBLIC 不同，此结构以传递给操作代码的字节数组的形式解析。

表 A.128 TCM2B_TEMPLATE 结构体定义

参数	类型	描述
size	UINT16	publicArea大小
buffer[size]{:sizeof(TCMT_PUBLIC)}	BYTE	公共结构数据

A.6.2 私钥数据结构

A.6.2.1 TCMU_SENSITIVE_COMPOSITE

TCMU_SENSITIVE_COMPOSITE 定义，见表 A.129。

表 A.129 TCMU_SENSITIVE_COMPOSITE 结构体定义

参数	类型	选择位	描述
ecc	TCM2B_ECC_PARAMETER	TCM2_ALG_ECC	私钥参数
bits	TCM2B_SENSITIVE_DATA	TCM2_ALG_KEYEDHASH	私钥数据
sym	TCM2B_SYM_KEY	TCM2_ALG_SYMCIPHER	对称密钥
any	TCM2B_PRIVATE_VENDOR_SPECIFIC	—	密钥存储的供应商特定大小

A.6.2.2 TCMT_SENSITIVE

授权值不应大于由对象的算法名称生成的摘要的大小，见表 A.130。种子值应为对象算法名称生成的摘要的大小。

表 A.130 TCMT_SENSITIVE 结构体定义

参数	类型	描述
sensitiveType	TCMI_ALG_PUBLIC	保密区域的标识符。 应与关联公共区域的类型参数相同
authValue	TCM2B_AUTH	用户授权数据。 授权值可以是零长度字符串
seedValue	TCM2B_DIGEST	对于父对象，可选的保护种子；对于其他对象，混淆值
[sensitiveType]sensitive	TCMU_SENSITIVE_COMPOSITE	特定于类型的私有数据

A.6.2.3 TCM2B_SENSITIVE

TCM2B_SENSITIVE 结构用作 TCM2_LoadExternal()中的参数，见表 A.131。它是未加密的敏感区域，但可使用参数加密对其进行加密。

表 A.131 TCM2B_SENSITIVE 结构体定义

参数	类型	描述
size	UINT16	保密数据大小
sensitiveArea	TCMT_SENSITIVE	未加密的保密区域

A.6.2.4 TCM2B_PRIVATE

在创建、加载和修改对象敏感区域的多个命令中，本结构用作参数，见表 A.132。

表 A.132 TCM2B_PRIVATE 结构体定义

参数	类型	描述
size	UINT16	私钥结构大小
buffer[size]{:sizeof(_PRIVATE)}	BYTE	加密的私钥区域

A.6.3 标识对象

A.6.3.1 TCMS_ID_OBJECT

TCMS_ID_OBJECT 定义，见表 A.133。

表 A.133 TCMS_ID_OBJECT 结构体定义

参数	类型	描述
integrityHMAC	TCM2B_DIGEST	使用目标 TCM 上存储密钥的算法的 HMAC
encIdentity	TCM2B_DIGEST	如果名称与引用的对象匹配，则返回保护信息。 所有的加密，包括大小字段

A.6.3.2 TCM2B_ID_OBJECT

TCM2B_ID_OBJECT 定义，见表 A.134。

表 A.134 TCM2B_ID_OBJECT 结构体定义

参数	类型	描述
size	UINT16	凭证结构的大小
credential[size]{:sizeof(TCMS_ID_OBJECT)}	BYTE	加密的凭证区域

A.7 NV存储结构

A.7.1 TCM2_NV_INDEX

TCM2_NV_INDEX 用于引用 NV 内存中定义的位置,见表 A.135。

表 A.135 TCM2_NV_INDEX 结构体定义

比特	名称	描述
23:00	index	NV位置的索引
31:24	RH_NV	指示NV索引范围的TCM2_HT_NV_INDEX的常量值

A.7.2 TCMA_NV

此结构允许 TCM 跟踪数据和操作 NV 索引的权限,见表 A.136。

平台物理控制(TCMA_NV_PPWRITE 和 TCMA_NV_PPREAD)和所有者控制(TCMA_NV_OWNERWRITE 和 TCMA_NV_OWNERREAD)使平台和所有者可使用平台授权或所有者授权访问 NV 索引,而不是使用索引的 authValue 或 authPolicy。

如果设置了 TCMA_NV_AUTHREAD,若提供了索引 authValue,则可读取索引。如果设置了 TCMA_NV_POLICYREAD,则可在满足索引 authPolicy 的情况下读取索引。

应至少设置一个 TCMA_NV_PPREAD、TCMA_NV_OWNERREAD、TCMA_NV_AUTHREAD 或 TCMA_NV_POLICYREAD。

如果设置了 TCMA_NV_AUTHWRITE,若提供了索引 authValue,则可写入索引。如果设置了 TCMA_NV_POLICYWRITE,则可在满足索引 authPolicy 的情况下写入索引。

应至少设置一个 TCMA_NV_PPWRITE、TCMA_NV_OWNERWRITE、TCMA_NV_AUTHWRITE 或 TCMA_NV_POLICYWRITE。

表 A.136 TCMA_NV 结构体定义

位	名称	描述
0	TCMA_NV_PPWRITE	SET(1):如果提供平台授权,则可写入索引数据。 CLEAR(0):不能通过平台授权来授权索引数据的写入
1	TCMA_NV_OWNERWRITE	SET(1):如果提供了所有者授权,则可写入索引数据。 CLEAR(0):在拥有所有者授权的情况下,无法授权索引数据的写入
2	TCMA_NV_AUTHWRITE	SET(1):如果提供了索引授权值,则可写入索引数据。 CLEAR(0):无法使用索引授权值,授权索引数据的写入
3	TCMA_NV_POLICYWRITE	SET(1):用于更改索引内容的授权,用户角色提供策略会话。 CLEAR(0):用于更改索引内容的授权,用户角色可以不是策略会话
4	TCMA_NV_COUNTER	SET(0):索引包含一个8比特字节的值,该值将用作计数器,并且只能用TCM2_NV_Increment()修改 CLEAR(0):索引不是计数器
5	TCMA_NV_BITS	SET(1):索引包含一个8比特字节的值,用作位字段,只能用TCM2_NV_SetBits()修改。 CLEAR(0):索引不是位字段

表 A.136 TCMA_NV 结构体定义（续）

位	名称	描述
6	TCMA_NV_EXTEND	SET(1):索引包含一个像PCR一样使用的摘要大小的值。只能使用TCM2_NV_Extend修改索引。extend将使用索引的nameAlg。 SET0):索引不是PCR
9:7	reserved	0
10	TCMA_NV_POLICY_DELETE	SET(1):除非使用TCM2 NV_UndefineSpaceSpecial()满足身份验证策略,否则不能删除索引。 CLEAR(0):可使用TCM2 NV_UndefineSpace()使用正确的平台或所有者授权删除索引
11	TCMA_NV_WRITELOCKED	SET(1):无法写入索引。 CLEAR(0):可写入索引
12	TCMA_NV_WRITEALL	SET(1):不允许部分写入索引数据。写入大小应与定义的空间大小匹配。 CLEAR(0):允许部分写入。如果索引的.dataSize大于实现的NV_MAX_BUFFER_SIZE,则应设置该比特位
13	TCMA_NV_WRITEDEFINE	SET(1):TCM2_NV_WriteLock()可用于防止进一步写入此位置。 CLEAR(0):如果TCMA NV_WRITE_STCLEAR也是CLEAR,则TCM2_NV_WriteLock()不会阻止后续写入
14	TCMA_NV_WRITE_STCLEAR	SET(1):TCM2_NV_WriteLock()可用于在下次TCM重置或TCM重新启动之前阻止进一步写入此位置。 CLEAR(0):如果TCMA_NV_WRITEDEFINE也是CLEAR,则TCM2-NV-WriteLock()不会阻止后续写入
15	TCMA_NV_GLOBALLOCK	SET(1):如果TCM2_NV_GlobalWriteLock()设置成功,TCMA_NV_WRITELOCKED被设置。 CLEAR(0):TCM2_NV_GlobalWriteLock()不影响在此索引处写入数据
16	TCMA_NV_PPREAD	SET(1):如果提供平台授权,则可读取索引数据。 CLEAR(0):平台授权无法授权读取索引数据
17	TCMA_NV_OWNERREAD	SET(1):如果提供了所有者授权,则可读取索引数据。 CLEAR(0):无法使用所有者授权授权读取索引数据
18	TCMA_NV_AUTHREAD	SET(1):如果提供了索引授权值,则可读取索引数据。 CLEAR(0):无法使用索引授权值,读取索引数据
19	TCMA_NV_POLICYREAD	SET(1):如果满足授权策略,则可读取索引数据。 CLEAR(0):不能使用索引身份验证策略,读取索引数据
24:20	reserved	0
25	TCMA_NV_NO_DA	SET(1):索引的授权失败不会影响DA逻辑,并且当TCM处于锁定模式时,索引的授权不会被阻止。 CLEAR(0):索引的授权失败将增加授权失败计数器,当TCM处于锁定模式时,不允许对此索引进行授权

表 A.136 TCMA_NV 结构体定义（续）

位	名称	描述
26	TCMA_NV_ORDERLY	SET(1):仅当TCM执行断电(TCM2_shutdown())时,应保存NV索引状态。 CLEAR(0):在更新索引的命令成功完成之后,要求NV Index state是持久的(NV update与update命令是同步的)
27	TCMA_NV_CLEAR_STCLEAR	SET(1):为索引写入的TCMA_NV_WRITTEN为CLEAR,通过TCM Reset或TCM Restart。 CLEAR(0):写入的TCMA_NV_WRITTEN未被TCM重新启动更改
28	TCMA_NV_READLOCKED	SET(1):在下次TCM重置或TCM重新启动之前,将阻止对索引的读取。 CLEAR(0):如果提供了适当的授权,则允许读取索引
29	TCMA_NV_WRITTEN	SET(1):索引已写入。 CLEAR(0):索引未写入
30	TCMA_NV_PLATFORMCREATE	SET(1):此索引可能未经平台授权定义,但未经所有者授权。 CLEAR(0):此索引可使用所有者授权未定义,但不能使用平台授权
31	TCMA_NV_READ_STCLEAR	SET(1):TCM2_NV_ReadLock()可用于为此索引设置TCMA_NV_READLOCKED。 CLEAR(0):TCM2_NV_ReadLock()对该索引没有影响

A.7.3 TCMS_NV_PUBLIC

此结构描述一个 NV 索引,见表 A.137。

表 A.137 TCMS_NV_PUBLIC 结构体定义

名称	类型	描述
nvIndex	TCMI_RH_NV_INDEX	数据区的句柄
nameAlg	TCMI_ALG_HASH	用于计算索引名称并用于授权策略的杂凑算法
attributes	TCMA_NV	索引属性
authPolicy	TCM2B_DIGEST	索引的可选访问策略。如果不存在授权策略,则注释应为空策略
dataSize{:MAX_NV_INDEX_SIZE}	UINT16	数据区域的大小
=TCM2_RC_SIZE	—	当请求的大小对于实现来说太大时返回的响应代码

A.7.4 TCM2B_NV_PUBLIC

在 TCM 接口上发送 TCMS_NV_PUBLIC 时使用此结构,见表 A.138。

表 A.138 TCM2B_NV_PUBLIC 结构体定义

名称	类型	描述
size=	UINT16	nvPublic大小 “=”表示由TCM来验证结构体的其余部分的长度，也就是缓冲区中用于成功解组结构体的字节数应与size相同
nvPublic	TCMS_NV_PUBLIC	公共区域

A.8 上下文数据

A.8.1 TCM2B_CONTEXT_SENSITIVE

TCM2B_CONTEXT_SENSITIVE 接口类型定义，见表 A.139。

表 A.139 TCM2B_CONTEXT_SENSITIVE 结构体定义

参数	类型	描述
size	UINT16	大小
buffer[size]{:MAX_CONTEXT_SIZE}	BYTE	敏感数据

A.8.2 TCMS_CONTEXT_DATA

TCMS_CONTEXT_DATA 接口类型定义，见表 A.140。

表 A.140 TCMS_CONTEXT_DATA 结构体定义

参数	类型	描述
Integrity	TCM2B_DIGEST	统一值
Encrypted	TCM2B_CONTEXT_SENSITIVE	敏感数据

A.8.3 TCM2B_CONTEXT_DATA

TCM2B_CONTEXT_DATA 接口类型定义，见表 A.141。

表 A.141 TCM2B_CONTEXT_DATA 结构体定义

参数	类型	描述
size	UINT16	大小
buffer[size]{:sizeof(TCMS_CONTEXT_DATA)}	BYTE	数据

A.8.4 TCMS_CONTEXT

TCMS_CONTEXT 接口类型定义，见表 A.142。

表 A.142 TCMS_CONTEXT 结构体定义

名称	类型	描述
sequence	UINT64	上下文的序号 注：永久对象上下文和会话上下文使用不同的计数器。
savedHandle	TCMI_DH_CONTEXT	会话，对象，序号句柄
hierarchy	TCMI_RH_HIERARCHY+	上下文等级
contextBlob	TCM2B_CONTEXT_DATA	上下文数据和完整HMAC

A.8.5 TCMS_CONTEXT

Context Handle 的值如表所示，见表 A.143。

表 A.143 Context Handle 值

值	描述
0x02xxxxxx	一个HMAC会话上下文
0x03xxxxxx	一个策略会话上下文
0x80000000	一个普通的transient对象
0x80000001	一个序列对象
0x80000002	一个设置了stClear属性的transient对象

A.8.6 TCMT_TK_CREATION

TCMT_TK_CREATION 接口类型定义，见表 A.144。

表 A.144 TCMT_TK_CREATION 结构体定义

参数	类型	描述
tag {TCM2_ST_CREATION}	TCM2_ST	ticket结构标识
=TCM2_RC_TAG	—	如果tag不是TCM2_ST_CREATION的错误返回值
hierarchy	TCMI_RH_HIERARCHY+	等级包含的名字
digest	TCM2B_DIGEST	可以是使用等级证明值HMAC产生

A.8.7 TCMT_TK_VERIFIED

TCMT_TK_VERIFIED 接口类型定义，见表 A.145。

表 A.145 TCMT_TK_VERIFIED 结构体定义

参数	类型	描述
tag {TCM2_ST_VERIFIED}	TCM2_ST	ticket结构标识
≠TCM2_RC_TAG	—	如果tag不是TCM2_ST_VERIFIED的错误返回值
hierarchy	TCMI_RH_HIERARCHY+	等级包含的密钥名字
digest	TCM2B_DIGEST	可以是使用等级证明值HMAC产生

A.8.8 TCMT_TK_AUTH

TCMT_TK_AUTH 接口类型定义，见表 A.146。

表 A.146 TCMT_TK_AUTH 结构体定义

参数	类型	描述
tag {TCM2_ST_AUTH_SIGNED, TCM2_ST_AUTH_SECRET}	TCM2_ST	ticket结构标识
≠TCM2_RC_TAG	—	如果tag不是TCM2_ST_AUTH的错误返回值
Hierarchy	TCMI_RH_HIERARCHY+	等级对象，用于产生ticket
Digest	TCM2B_DIGEST	可以是使用等级证明值HMAC产生

A.8.9 TCMT_TK_HASHCHECK

TCMT_TK_HASHCHECK 接口类型定义，见表 A.147。

表 A.147 TCMT_TK_HASHCHECK 结构体定义

参数	类型	描述
tag {TCM2_ST_HASHCHECK}	TCM2_ST	ticket结构标识
≠TCM2_RC_TAG	—	当不是TCM2_ST_HASHCHECK的时候返回码
hierarchy	TCMI_RH_HIERARCHY+	等级
digest	TCM2B_DIGEST	可以是使用等级证明值HMAC产生

A.8.10 TCMI_ALG_PUBLIC

TCMI_ALG_PUBLIC 接口类型定义，见表 A.148。

表 A.148 TCMI_ALG_PUBLIC (TCM2_ALG_ID)类型定义

值	描述
TCM2_ALG_!ALG.o	所有对象类型
=TCM2_RC_TYPE	当公共类型不支持的时候的返回值

A.8.11 TCMT_PUBLIC

TCMT_PUBLIC 接口类型定义,见表 A.149。

表 A.149 TCMT_PUBLIC 结构体定义

参数	类型	描述
type	TCMI_ALG_PUBLIC	对象分配的算法
nameAlg	+TCMI_ALG_HASH	用于计算名称对象的算法
objectAttributes	TCMA_OBJECT	属性,以及类型,确定对此的操作对象
authPolicy	TCM2B_DIGEST	使用密钥的可选策略 使用对象的nameAlg来计算策略 如果没有授权策略,可为空的buffer
[type]parameters	TCMU_PUBLIC_PARMS	详细算法或者结构体
[type]unique	TCMU_PUBLIC_ID	结构体的统一标识 一个非对称密钥,这个应是公钥
注:“+”表明实例可有 TCMT_PUBLIC,一个“+”表明 nameAlg 可为 TCM2_ALG_NULL。		

A.8.12 TCM2B_PUBLIC

TCM2B_PUBLIC 接口类型定义,见表 A.150。

表 A.150 TCM2B_PUBLIC 结构体定义

参数	类型	描述
size=	UINT16	publicArea大小 “=”会强制 TCM尝试解析 TCMT_PUBLIC和检查,解析的大小与值大小的匹配
publicArea	+TCMT_PUBLIC	公共区域 “+”表明调用者可指定 nameAlg使用 TCM2_ALG_NULL

附 录 B
（资料性）
可信密码模块证实实例

B.1 概述

可信密码模块协议可行性证实主要是证明本协议描述的协议和机制是可实现的。本附录内容仅用于评估 TCM 实现实例与第 7 章接口要求的符合性。本附录给出的数据都是实例，实际中接口的输入是上层根据业务需求编排输入内容，平台按输入的内容操作返回相应的输出。

8.1 是符合性实现实例。B.2～B.20 使用与 8.1 相同的测试方法获得的实测结果，因为某些命令接口的输入输出内容具有随机性，所以基于不同的 TCM 实例，会导致最终测试结果不同。

注：更多与接口符合性测试规范的测试方法可参考GM/T 0013来执行。可信密码模块实证验证主要是测试不同厂商生产的可信密码模块实现的功能和接口正确性，确保这些可信密码模块产品都以相同的方式接收和运行命令，与厂商无关的命令可直接通过示例命令向量进行测试，验证每条命令的参数格式、结构解析以及操作执行输入输出的正确性。

B.2 启动命令输入输出实例

B.2.1 TCM2_Startup

输入：
80010000000c000001440000
输出：
80010000000a00000000

B.2.2 TCM2_ShutDown

输入：
80010000000c000001450000
输出：
80010000000a00000000

B.3 检测命令输入输出实例

B.3.1 TCM2_SelfTest

输入：
80010000000B0000014301
输出：
80010000000A00000000

B.3.2 TCM2_IncrementalSelfTest

输入：
8001000000100000014200000001000 5
输出：
8001000000380000000000000015000b001200130023000600 1b000a000800100014001500160017001

80019001a001c0020002200250043

B.3.3 TCM2_GetTestResult

输入：

80010000000A0000017C

输出：

80010000000A00000000

注：基于不同厂商的自定义输出可出现不同的结果。

B.4 会话命令输入输出实例

B.4.1 TCM2_StartAuthSession

输入：

80010000003B00000176400000074000000700200E16C8006E268FE59A7C021DECF067023BADC0877D235E9D4E3230389CC14CE500000000100012

输出：

8001000000300000000002000000020A47796438DF66C348A0051452A800726C7ADF798AA749DB27C915FCD31A3BE6F

B.4.2 TCM2_PolicyRestart

输入：

80010000000e0000018003000001

输出：

80010000000a00000000

B.5 对象命令输入输出实例

B.5.1 TCM2_Create

输入：

8002000000410000015380000000000000c4000000900000000373746f00070003616573000000120025000b00060472000000130080004300000000000000

输出：

8002000001a60000000000000193008e002050e5def77e79e2c3dd286afee5064d8696fe97cf1690ba020b075d0542330570010f3898482d224635b2d877b13fd2f22870e62627457dd506fd251ee1be9c921512bf6672a667f742a68cf1f6e96e3cf2e982edd33e819b09fbf43145cd0d1db7a8edf626c6cdd349e68f9e9362dd178c8903a443ca0be71a68e5c966fea719fd7a3d51512deb08a5d2100320025001200060472000000010080004300 20c46e90d2d93ab5d0a6c1e2d5da59db14caf5a1592ef4672995ebcce68f87d62e00730000000000020e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b85501000b0022000bd40b31438d83e0bb7599d9f6ace904d47bf6ed45ec1aab211fa7b1b2caf99680022000ba94c86b883dad5dd8bf6bdade6d53c9e9aa77b9c0da42cf26d405598ad516d000000203cdebe56e4a75f5148a51e09ef578987358f3f2ea333fd27b96748c27db8a8780214000000c003096aca236b4c42b3f1ca5498998617b53457c1c06ad75cc73d18171c7674667209d6b6dde2c63a9a0438b4a5a1c18020000010000

B.5.2 TCM2_Load

输入：

8002000000e2000001578000000000000000c4000000900000000000373746f008e002050e5def77e79e2c3d286afee5064d8696fe97cf1690ba020eb075d0542330570010f3898482d224635b2d877b13fd2f22870e6627457dd506fd251ee1be9c921512bf66672a667f742a68cf1f6e96e3cf2e982edd33e819b09fbf43145c0d1db7a8edf626c6cdd349e68f9e93642dd178c8903a443ca0be71a68e5c966fea719fd7a3d51512deb085d2100320025001200060472000000130080004300 20c46e90d2d93ab5d0a6c1e2d5da59db14caf5a1592f4672995ebcce68f87d62e

输出：

80020000003b0000000080000001000000240022000b26500a4b538c5e0e37b7169e71714e6d5f6f179e6221dd9fc855b115d76d4880000010000

B.5.3 TCM2_LoadExternal

输入：

80010000003c000001670018002500000000001011223344556677889900aabbccddeeff0012002500100060440000000130080004300004000007

输出：

8001000000100000000080000000000

B.5.4 TCM2_ReadPublic

输入：

80010000000e000001738100000 1

输出：

8001000000ae00000000005a0023001200030472000000130080004300100020001000204 0d80c9258ad28650c9150e01f364f18a76cd0d72c431ab729a34ad3c2cbbb300205af388e9f3265867c138985b6301552d3669f1992afac769a9a61284ebc122200220012913e07579880abd0761c17031355d7ee894b80aefccb1940d2a1f55e172279800220012ab249e3bf27ba4dab2b11e50e9ce33d3b82914277868c5e9e4555fec8ce95e6

B.5.5 TCM2_ActivateCredential

输入：

8002000001120000014780000001800000000000007a03000000001470e720083515b2d24edc2f1cf6c09e3574ec9d1a0100140e69f47c074f3d2a6a2aba55270a2b502056ff8602000001001058337 7ff3910db67088b5d5b51a6bca2010030e334cb7bdae866652b2b6226ac4bff1ced5ef55f1aa10246182f88f2665711b5cf41b983d55f1798d817c0c1704c5c8f003a0020f130ca05061c034dce714d2ba98f080743b515c8306567eea6c75cc1c06d8b7f4220eae68c2d0d247f23d86024108c4a5ff94d5adfabfb3900440020a3bd0ef7a4e0b2330ebb483edabac31644e32caf49bc80fae02c0122db0f80a50020bf1d456d775ec0bded9963825fae16743dc8c4d2d16c9759cf99dce3a296b1f1

输出：

8002000000980000000000000180016ce02abb50fc64bf012b8bb34f46b7550ed22c86ab6920014985ef463f9855f7495f52ce4fc35a55337f79aa20100148e80d533a6e36e633a14c36f6803981f0ff3fef40010632d

a6573c4a696681a84c213c0d414e01003014038e5b50a15ba0c33b24514ea8573bd5b86921b0c06df04cf0a3e029763a420ee5a3212b5bfe36a5914467f2b52346

B.5.6 TCM2_MakeCredential

输入：

80010000003e00000168800000000016ce02abb50fc64bf012b8bb34f46b7550ed22c86ab69200160004a091c1963d4933ae7c428ec1ead33922c3137a6e

输出：

80010000008c00000000003a002015bf21e556e963f613739af89c3734a09bd169319925f5d4edf80dc39049671fc0348f1301f3e30c5acea821640b2e235b0494cb16eb11600044002077af06c47d17e3aee5643f36e2d4c61ff61ba6298cd0c76601cc2fccf1d4e0f900208ccdf0110242d2d276c0096912bcc6f9795c1256b09513de22311cd4b4a609d0

B.5.7 TCM2_Unseal

输入：

8002000000410000015e800000020000002f020000000012514a2da8083bc185b0ffc302bc288470b54d0100148934392db1c3f180ecb320c417b92a14c4dce698

输出：

80020000005c00000000000000023002152a0853b718cc6a40291ddbbbd644466acf5489c9a974c4adab3530758dcfad9d10012d535db476113843d9f612ddaba99b53772f401001450a90acc42242ce4ce6a8e41d3e7e20bc587d24e

B.5.8 TCM2_ObjectChangeAuth

输入：

8002000000270000015080000001800000000000000C4000000900000000037369670003787878

输出：

8002000000930000000000000080007e0020d70d15149458b3e6b401e0c5ebff33a6b85558f23bb58492db402d30c42c8b7900102dd9ab9187fc8bf58c8c330c5bb9e9f638aacd84e0bc641e856cbb91e216f72f7b7c129b5bd1a24192726b70caf0b824ec9afb4c827fc913aa4a79e6703fda6ab44ed3c7a1819dab3a200aa19e66a029abc5893af935e1501a5c0000010000

B.6 复制命令输入输出实例

B.6.1 TCM2_Duplicate

输入：

80020000004b0000014b8000000040000007000000310300000000014db51a06603da2527c9e4944b3e9de172a6f32a570100149c0c11339317963fbb5bfbe01e5067a86346c05600000010

输出：

8002000000930000000000000058000000520050002500142700000000000000000000000000000000000000000148853055328187183 3c833f4335469636a961ebd000206245febe4e72fb997dc3e31f95e04f1cc4a48a728cc3e8dc6c73fb0690a4147900000014b9186cdead30fc03b97a81e54582dc5e5d74ee860100144fff9645c4ca30d45bf30eafe64c9367e82cfcf6

B.6.2 TCM2_Rewrap

输入：

8002000002750000015280000000080000001000000490200000000102347519658498b66f99a3f610a8f
90470100305e0da0606c5b805be41d7971c59bc2aa4d7e15f7ce24309d825f53019a11a90faa057535d28c
76dea461db2fafeec54c00ee00201b4148c5e189297dd2c198cc4a0f866ef4671fe3bd2df7811c6e324866d7
09719c1d4d1018d9331f2a5da4c2d04e88fa5b51281e699e2b5a665a30a47746d5cc4e7dc9efe3a73d9e05
03259b1820a90fb34b59a0c41b97d5bf614007939f4a6fabdf9a1d0fd2cc144210e901c77cfe9ff978ad7799
8fe1e67415247e8b592b487fb835a1945462e4e1118cc50e0e07f54646ade734898103a9c5a8a89f6050b7
25f8ad8fc4f8ce0ed0227b82a8d7a3e330f40841813e1fc6891261d526fa3f9d8d4dd888ee
afc574753c88393c7a66989b77cd583a01c7ce5ca6125e4d66325ef5b70c721079c0605e18f5130022000b
09d610211ffb91480b5142df09758786a4800c6d3bb1e7f987ae64943a4b2208010090a28b0345d11c307c
8fd2844aec2fec7450a7c088a863cde2764c693e670e008ad8d7ce246ee1a8fff88c26c7e5750f2c1e607c73
858d42400259362dd9be03f70277620c8708abb21b04331c6b3f3e84edead5917b16ab1955c1f64c06f02a
c31872255a01977b2dcfb207c38f6e4998f0725871bb040ef124291213d9950f8f1669e20387bdf375ffe33
7a64fe3cb7690a7499fcc479f306d34b15ab222e17f469bcbf1d5ac4dda7b5b200d96d29dde04f16d3bbe53
153c7e2d946e67258ae74a5dc9e579c02256e2b295a1db23ced4c57ea2cb95d767394c7c050667eebc38e1
ca07bdb69391fb5a85d603d543cf416b65b23c1bfcf38fa225b9f1a6ea52

输出：

8002000002450000000000000001f200ee0020d835f6ea54976ed9d77e199512e7b0a9f25b4bcab2f3a049cb
7908c0826261864f913d4c2d625e8966c7190e4e44d7421e201145a913e4c42e6615d8e7ef02dc752f31f0
457b793ce86425f893b5ce0f13f44b3d9221b3959310c8d024b6b8f64627e23ce74b36c8e7af749032a903
4acc62a61e493e7404ad1708f3587787693c7d80d4437e2572aac306592f1bcafae404d89632afe20a47232
d8442076e3407de140c1cdf0c949136554e0eb2611b121aed5e22feb36d64e5cc0263747e1f9fcdee57f471
4cd73c0e00b64583e6af9e08fae7de1fa19bda3c4d24fe0b9647647720912d095d4ba85f465d01006b49a4b
ddc20e34826147de0f889dbf050bc43bb767054218aa791913e1412c42e00a02ee7676f315c2275670b2b8
26b3777837da91c69f70a7411a5ec4e7f0b390bc33f21f1f20c087f453df944d0b375dda50d4f7e9c9f2923e
8c8123a21edc3d7e4b8bcda1d40df953f5d6f16163a7d8265c8c0185d9e7a171ad145eb5c75c9018473ed
bfb4be1cc4e71d77a3972b9d37f3609e3205aa9a90d9df16513de56e48140312a7849b4fc2dd8c0c294a7d
95d5cbfeaa553bb3ba4db701b3441654cc22fb32092ad4ec43aa728cc6531ed34539588c1230bab91f1309
72a1c274194204d7e37b74bea7e67ded708700f068a1df8d1365a9a3b29e188c4e39c0e6150010e3bb840
b6b90879d956a7a1ff09155b301003093bbc4347f57a1649f1f4b00d251a0d70e38af92ac60e4edb24675b2
5ef7b93d75eb8ce179d64363a7e803213908819e

B.6.3 TCM2_Import

输入：

80020000014e000001568000000000000000c4000000900000000373746f000000760023001200040460
0020bf1d86ffbfab26d0170e9c700b14db60ad438b39331a10312c9fd50c3ab7b591001000100020001000
20abcfa0f10cc8c55572fc6e22d8793268e22b028e4e1c4f0bf96d34e409489bef00200fb6a52dd52007e3c3
db30de61bc4a226c29db99384a7bfe42811b2a0f67c26e006c00203ad3db754b8a7624af9b64e03b9fc7d6
7ac8c537ed706010e7a71ffede1112edcfe9f114ddf056cf2ab8a4216297ae201e037d0770c2c7f996f785e7
8dd971a67cbef881162bf709269378f2fd08351a2bb5d2b161dad8b91c8b69dd8a70f778c033f92f3fac142

b534100440020a7a5160912a68176b236533c2ddeaebd9c89ec930a3b6bcbf2bb9150685df17700200018
3e65f52f7b6e08ffe693415bc5c307877c575a64ac561dc02d9de02be88a0010

输出：

8002000000930000000000000080007e00206abb6c6c334380929715adf3f1dbf3129e0336de0bba73bd6
8568c111d4d57ac00105b6d90c6f7f98d5dc25dc471a23979899524d6666851c2040ee89736d919eb38d4
2c8c48da30e9214e9682572670cdc4c8d9bd4663af156c59a87ceaface9cada8b6a24185b795007d019abc8
c69d9d92152013c9649ab8411c50000010000

B.7 非对称算法命令输入输出实例

B.7.1 TCM2_ZGen_2Phase

输入：

8002000000ab0000018d8000000100000009400000090000000000000440020b9e5158aeed6909b8a8400
48fb3ecb2c2b4356c5678228ae9e70d6c3bf2728520020a8ce7d2d788ec2016f9aac7d9ef51a77b4dfc4844f
d686dc936573cdb0080cb100440020256b295934239504d8fa0867c088d82ac4f596637f24ace24c8b35b0
1e0ea2990020f1c79537fabeed0a0d11e3622a31a7a1ca4523725d2978b34a097d3e40b129d300190002

输出：

80020000009f0000000000000008c00440020398ad6c13c390332ef63165b63dbafded9d7524c14d1e6c0d
319f6fe5db081750020c17be8947a3c884d61209115ac3ac20bfcb8f6da774e8fa76be1b425ee81771b0044
00203e4163ed3fdabac3efb62f529165965e98c9473a15782f1143fec4a07edf497d0020b63669e8d090c27
f8f3fe9888952d3f679a5bd94b50bcb61a79c4879c95a36b20000010000

B.7.2 TCM2_ECC_Encrypt

输入：

8001000000170000019980000001000361616100210012

输出：

8001000000770000000000440020f20589ee1e993cdfb878b4a0545262088b69d6c63cc9e98165c491ee3
e513e9100206e3368a8a24e945d8ee636e134515e78ee72f86c99bb611e33caf56cf6de043700039b45d60
02011f4507d5c6d47c3843de7c3add54b46ec290854896287062bbac3eb46ed7d8e

B.7.3 TCM2_ECC_Decrypt

输入：

8002000000cc0000019a8000000100000049020000000203f1488207c1b9da88bae88e6230eba6f62b56
c64685f3ab6a69116bc7f110606010020fdfafd7ac65b33fbd5b4b8f3c265c0795b037c8bf2c888ccf64f7d70
e884039200440020f20589ee1e993cdfb878b4a0545262088b69d6c63cc9e98165c491ee3e513e9100206
e3368a8a24e945d8ee636e134515e78ee72f86c99bb611e33caf56cf6de043700039b45d6002011f4507d5
c6d47c3843de7c3add54b46ec290854896287062bbac3eb46ed7d8e00210012

输出：

8002000000580000000000000000500036161610020d72a4c8958de150396e02b5daaea8fa208145ae3650
8ab8b9beddac003b428d60100208321df439b90898f9ac4f205d6fc9aeba3b9154d6cb737a437f4e447bc13
ae5a

B.8 对称算法命令输入输出实例

B.8.1 TCM2_EncryptDecrypt

输入：

8002000000450000019380000001000000c4000000900000000036165730010313233343536373839303132333435360000100010000000000000000000000000000000000000

输出：

8002000000370000000000000240010701fed12e32bf407ccc7eb1a758a9c5e0010701fed12e32bf407ccc7eb1a758a9c5e0000010000

B.9 随机数发生器命令输入输出实例

B.9.1 TCM2_GetRandom

输入：

80010000000c0000017b0020

输出

80010000002c000000000020bc57834701a0a776db6a1a53ef5800a4985cd1d44dc7921ab3c91feaf548e28d

B.10 HASH/HMAC命令输入输出实例

B.10.1 TCM2_Hash

输入：

8001000000160000017d000431313232001240000007

输出：

8001000000340000000000202322a919c5c7993acbac4f1f6025b3d226789c5f835d110bbd251fb6937165d18024400000070000

B.10.2 TCM2_HMAC

输入：

8002000000320000015580000001000000c4000000900000000036b686b00103132333435363738393031323333435360012

输出：

8002000000350000000000000000220020dedd8ffe59fb8686a877b1ff4880fdb7208093e1d5129158c1fa941c3f4092f30000010000

B.10.3 TCM2_HMAC_Start

输入：

8002000000250000015b800000010000000c4000000900000000036b686b00036161610012

输出：

8800200000017000000008000000200000000000000100000010000000a0000018a

B.10.4 TCM2_HashSequenceStart

输入：

80010000001100000186000361616100 12

输出：

80010000000e0000000080000001

B.10.5 TCM2_SequenceUpdate

输入：

8002000000230000015c800000010000000c4000000090000000003616161000361616 1

输出：

8002000000130000000000000000000000010000

B.10.6 TCM2_SequenceComplete

输入：

8002000000300000018500000010800000000000001540000009000000000040000009000000000036161610003616161

输出：

8002000000a8000000000000000900000000000400047e240de74fb1ed08fa08d38063f6a6a91462a815000b9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0000c8e07e5bdd64aa37536c1f257a6b44963cc327b7d7dcb2cb47a22073d33414462bfa184487cf372ee0a19dfc83f8336d800128d83c7af17f544dffb989f53cd6aafdc2eda6ca5ea7fef3dd7b2f0ee8230660d00000100000000010000

B.11 证书命令输入输出实例

B.11.1 TCM2_Certify

输入：

8002000000370000014880000001800000010000001840000009000000000373696740000009000000000037369670003616161001800 12

输出：

8002000000f2000000000000000da0090ff5443478017002200129 26f08b4fc4595ddaebd4c8d47e06fcbce3a19715f33b971df52c0ed22a764c8000361616100000000000000d3500000014000000000000007003700000000022001275643bd4e614bc250d2c62c11b4ada1b271ca25c16ba9673a597bcc1693fc28c00220012926f08b4fc4595ddaebd4c8d47e06fcbce3a19715f33b971df52c0ed22a764c8001800120020316c87d2d34aa0c277f95ba5c0b5e5eca70b437f52c3a36f56cc65f92ffafbb0002000042ccc6689d6dae2b15264dccc214d0d4f655e1cc3e06903a0d9aba47d7dcd00000100000000010000

B.11.2 TCM2_CertifyCreation

输入：

8002000000820000014a8000000180000000000000c4000000090000000003736967000000020c1f0fb18e8da773fcb980f440d847776171ad9411005855f0bee536bfee2f37c0018001280214000000c0030c5020a810173587771f7c0a7481713d39e72ffc63d4940339aa35462ffc84dcb2f74a2af6373c542a26bc4ad38cc7001

输出：

8002000000e800000000000000d5008bff544347801a00220012926f08b4fc4595ddaebd4c8d47e06fcbce3a19715f33b971df52c0ed22a764c8000000000000000003007000000140000000000000007003700000000022001292e947c2e7e0e8862176d047f870cf9396a443cd1cab896679d0c5604923cc040020c1f0fb18e8da773fcb980f440d847776171ad9411005855f0bee536bfee2f37c001800120020c0e01ac76e2875400b6d073875e9726325ad7667591b00e5ab4ea1dc5366f51800200637be8be76c01319df647b873bc4468b2b026465c6c875ce5d3c0bca3127b1b0000010000

B.11.3 TCM2_Quote

输入：

8002000000310000015880000001000000c4000000900000000037369670003616161001800120000000100120301 00

输出：

8002000000d100000000000000be0074ff5443478018002200129 26f08b4fc4595ddaebd4c8d47e06fcbce3a19715f33b971df52c0ed22a764c8000361616100000000000000e9b0000001400000000000007003700000000000000100120301 00000020e0bab8f4d8172ba245190d13c94117e93b82166c25b2b69883350c192c9051400018001200203 8fd1235dad3ebb49801ce50d36e2d7d896a4bc8242e1826f951d101d163cc3d0020d70b980eb7ab5bb320a8632113e3fd282a179d2ff778382454f9b3c29028b6df0000010000

B.11.4 TCM2_GetTime

输入：

8002000000340000014c4000000b8000000100000015400000090000000000400000090000000000373696700036161610018001 2

输出：

8002000000cb00000000000000b30069ff5443478019002200129 26f08b4fc4595ddaebd4c8d47e06fcbce3a19715f33b971df52c0ed22a764c8000361616100000000000000fc5000000140000000000000070037000000000000000000000fe50000000000000fe5000000140000000000000700370000000000001800120020b063f1378a6cdf8f96086b563fda3c94ad197a9e7f864257b5e19b5bf419de8800202d901b65aa998c7846b99a182747a71b1593786da755820e5a0c3feb1592f30e00000100000000010000

B.12 临时EC命令输入输出实例

B.12.1 TCM2_EC_Ephemeral( )

输入：

80010000000c0000018e0020

输出：

8001000000520000000000440020dfb0fa2102a7898baa8888894a8ca47ce96b607fb6b218c09b7eaf01eEd267fc0020a1972f7f593d00ae7965e428039af028829805fa8604ec25180dcf6a38ca98290000

B.13 签名及签名验证命令输入输出实例

B.13.1 TCM2_VerifySignature

输入：

8001000000780000017780000000200208d83c7af17f544dffb989f53cd6aafdc2eda6ca5ea7fef3dd7b2f
ee8230660d001800120020c88a119c59d4699de670deef56d8ad1e947543bb5c0a8f28d113e6a762f22e40
020eab5fda180a57c25c7f68d2f30657cce14aafe41306313de3b75105c2ee34325

输出：

8001000000420000000080224000000c00306ece6df3e6ad1303292656aea5a18a2d3cc67ee
bfd8c81b2b9a9f4d8b4487900730d4bd020c04d448bfa8a2b1bbb15b

B.13.2 TCM2_Sign

输入：

80020000004c0000015d800000020000000c4000000900000000003736967002 08d83c7af17f544dffb989f
53cd6aafdc2eda6ca5ca7fef3dd7b2f0cc8230660d001800128024400000070000

输出：

80020000005b0000000000000004800180012002 0c88a119c59d4699de670deef56d8ad1e947543bb5c0a
8f28d113e6a762f22e410020eab5fda180a57c25c7f68d2f30657cce14aafe41306313de3b75105c2ee3432
50000010000

B.14 度量命令输入输出实例

B.14.1 TCM2_PCR_Extend

输入：

8002000000410000018200000010000000094000000900000000000000000100126161610000000000000
0000000000000000000000000000000000000000000000000

输出：

80020000001300000000000000000000010000

B.14.2 TCM2_PCR_Read

输入：

8001000000140000017e00000001001203020000

输出：

80010000003E000000000000000140000000100120302000000000000100200000000000000000000000000
0000000000000000000000000000000000000000000000

B.14.3 TCM2_PCR_Reset

输入：

80020000001b0000013d000000100000000940000090000000000

输出：

80020000001300000000000000000000010000

B.15 增强授权命令输入输出实例

B.15.1 TCM2_PolicySigned

输入：

8001000000640000016080000001030000000000000000000000000000001b0012002007e69b5e1468a13

4787bfc59453a33f26603ae216a4cb86f974462312ffd7a900206d1d4fe316fe5a07a3ea6ffdf0519545a6497
5facd23dd8a56d0b15977c93ff
输出：
8001000000140000000000000802540000007000 0

B.15.2 TCM2_PolicySecret

输入：
8002000000290000001514000000C0300000000000000940000009000000000000000000000000000000000
输出：
80020000001D000000000000000A0000802340000007000000000010000

B.15.3 TCM2_PolicyTicket

输入：
80010000005c000001720300000000008000000000003c539000000000001600044234c24fc1b9de6693a6
2453417d2734d7538f6f8025400000070020cab2d84b6838485c04f9cb00cb2deb28828c055b7338173c6
1c1f08dfd1290a6
输出：
80010000000a00000000

B.15.4 TCM2_PolicyOR

输入：
8001000000560000017103000000000000020020f574f53bf73c4353696b5377162ee6ee7c7706c24309d
cbb2bb842304110b9fc00200d353c2fa24b68402b69b6cf390050c227d577a6f7e9459938308a172551f03
c
输出：
80010000000a00000000

B.15.5 TCM2_PolicyPCR

输入：
80010000001a0000017f0300000000000000001001203000000
输出：
80010000000a00000000

B.15.6 TCM2_PolicyCommandCode

输入：
8001000000120000016c0300000000000147
输出：
80010000000a00000000

B.15.7 TCM2_PolicyPhysicalPresence

输入：
80010000000e0000018703000000

输出：

80010000000a00000000

B.15.8 TCM2_PolicyCpHash

输入：

8001000000240000016e030000000014b5f919bbc01f0ebad02010169a67a8c158ec12f3

输出：

80010000000a00000000

B.15.9 TCM2_PolicyAuthValue

输入：

80010000000e0000016b03000001

输出：

80010000000a00000000

B.15.10 TCM2_PolicyGetDigest

输入：

80010000000e0000018903000001

输出：

80010000002c000000000020b6db9d60583fb1e996f8060fa0534495af3a55d3e0110fba978b7309aa6b38b4

B.16 分层命令输入输出实例

B.16.1 TCM2_CreatePrimary

输入：

8002000000660000001314000000c00000009400000090000000000007000373746f0000003a00230012000304720020000000000000000000000000000000000000000000000000000000000000000000000000000013008000430010002000100000000000000000000

输出：

80020000014a000000008000000000000133007a00230012000304720020000000000000000000000000000000000000000000000000000000000000000000000000000013008000430010002000100020258165dedb15d5e02eefe2d62ce36be31c010534cdc2ae8b69fed582ed1b420c00201ee4d8caca62ca868cc2dcfa83831a53405a0af5354022644bb3c6eafa803d1c00370000000000201ab21d8355cfa17f8e61194831e81a8f22bec8c728fefb747ed035eb5082aa2b01001000044000000c00044000000c00000020c1f0fb18e8da773fcb980f440d847776171ad9411005855f0bee536bfee2f37c80214000000c0030c5020a810173587771f7c0a7481713d39c72ffc63d4940339aa35462ffc84dcb2f74a2af6373c542a26bc4ad38cc70010022001292c947c2c7c0c8862176d047f870cf9396a443cd1cab896679d0c5604923cc040000010000

B.16.2 TCM2_HierarchyControl

输入：

8002000000200000001214000000c00000009400000090000000000004000000101

输出：

80020000001300000000000000000000010000

B.16.3 TCM2_SetPrimaryPolicy

输入：

80020000001f0000012e4000000c00000009400000090000000000000000010

输出：

80020000001300000000000000000000010000

B.16.4 TCM2_Clear

输入：

80020000001b000001264000000c00000009400000090000000000

输出：

80020000001300000000000000000000010000

B.16.5 TCM2_ClearControl

输入：

80020000001c000001274000000c00000009400000090000000000001

输出：

80020000001300000000000000000000010000

B.16.6 TCM2_HierarchyChangeAuth

输入：

80020000002000000129400000 0c00000009400000090000000000003707070

输出：

80020000001300000000000000000000010000

B.17 字典攻击命令输入输出实例

B.17.1 TCM2_DictionaryAttackLockReset

输入：

80020000001b000001394000000a00000009400000090000010000

输出：

80020000001300000000000000000000010000

B.17.2 TCM2_DictionaryAttackParameters

输入：

80020000002700000 13A4000000A00000009400000090000000000000000010000000000000001

输出：

80020000001300000000000000000000010000

B.18 管理功能命令输入输出实例

B.18.1 TCM2_PP_Commands

输入：

8002000000530000012d4000000c00000029020000000020D1B308C513801200C14E9095484EC8B5CE1E62C8D19E8498DE39656DB9A884E8010000000000020000014300000144000000100000145

输出：

8002000000530000000000000000002075346CA0E70B02D3F76F8CD1FEC759600C498E61D4C3AAC622BF9EE5EE5F06CC0100206D017910643FA7BA3B3CC4A28FCD9415B5E41D43A3B169D08AE20DE4188679DD

B.19 上下文管理命令输入输出实例

B.19.1 TCM2_ContextSave

输入：

80010000000e0000016280000000

输出：

8001000004560000000000000000000000000018000000004000000b043a00305cc18518cf47835b885ef5194a840aa491df0170446326614a424b6e74a57822481ed86c6d657e4f7c228a4ea16f66c17a119a24a4c9e02885a4e035d43cd1e9614ff9bc5105c674573b3fe4ed066c2827d605095100ccb8d7a9d187fd5419a2f2f984fe8cd19c7d6582747e7f0a5155a50009bf5678b0f23a2bb63abef1d6babd0592fcc44924f524e9654ff32c58cab03d133181078f1b7ed2cc170771310841561f9ece96c0b926814519286 07d63b71d38dbe6679d08f65d4347cfd10c8d5232ff7b4b9555f9742b35e850ced308fe4b06669cf2096ac298b28a04d45c0f44e5fa3f88da1b4a188988d01e709b9483c531ad96d84ed4b4c46f3bcdbbdf37a76f830b5716984365c2787ef1a61163fa328470ff03b03848746ad8925ae005521eddb082f15a2d9cf94092a438841446a3e3bc00c7c421e5c81b48b5d8b37a1f1370b7064f9b8603677f1dd6426321a9ffcc4590aad76f68b1dfd2c73094be9cafbe737e7d234424c06af151662e48af24569b9cb93771ab6227039a453534ad034d23d1efefad8487b43abe4b1e71a4d4265e2a4b093406634a0e8e24bb1cc4024ada401b4f3660b271443981c914dbf7a06869a727da3faae84799d2cc0fc5f16f110f98d9fb2531b9c2ae53c96689092919d8be80e8ddaaa5bad2dedaa4db7378bf5fef68af6df5ffd5736b0284685dc56fd06bc1b25312e4a9d0a57d62da93be67656a35fb02d3abac1cdd4bf1c564d813951105843cf6c207f8f2b3a466ffccbb2cc3dc633b1aea231505b4b3a7314cf197bdd1388e94fc1d186ad44410fcf1833949474cf62ca432a970bfecdfe9382409da64ae308dadfe300f7deffe908023f30ccf35b0d185c4af4f25362654feeaea9ed5616b2d8ebae7462c39f363be41b686c31f9e1377bac296365379d614d87dd487985fdae96e6ca82a5d9afda89b5b58dd3d4f1c078a6e785f43043d1abea6030b140df986f4ef74fb9d7a77d30e34bdf14ec3998dbb8669c70dcefd49040dd6a6aa814bf01911a578d60956b2426e14245aa-beb6e05cd6cf237b8bf78b59e68436cd2cfffaae69f2b7b753972dc24233108e2f00c8ab47b87d4e8d68121970759cedf721ba93f0d753e47da27f126af895d73481922d7b52e1765db06e78fdf4ffc935eedc36d3e27b389f9c87f77314a953a45aaa9cd62efccdaee5c10c8db6d063c80b8542a0588614700f7d2b1062515d68a7339c0846c73fa4c832745b3d750c68553375e5fafecca31b803c28b134178ac7bda7dfc2f1e171a6978c5dd6d03aeb38eed9ec08ecb3559d62602fb04927f9b16c5a19ea81d430c05d1e57b4067afb63e47e27e9da105b2f700643c297cb23c442be0e76b2a67c5d7492388915fb401efafec290ab2304dc75490844e2d98b34b393affb0dc8dd5d731fdeac759a5abbe894555d693010766a0485b6aada7c59a544bcd1d6e5539dd2954f8499cbb

591a851917f3c1aeeba233c6fc5277b31c42456e1879e23acdf0fef5fd928d

B.19.2 TCM2_ContextLoad

输入：

8001000004560000016100000000000000018000000004000000b043a00305cc18518ef47835b885ef519
4a840aa491df0170446326614a424b6e74a57822481ed86c6d657e4f7c228a4ea16f66c17a119a24a4c9e0
2885a4e035d43cd1e9614ff9be5105c674573b3fe4ed066c2827d605095100ccb8d7a9d187fd5419a2f2f98
4fe8cd19c7d6582747e7f0a5155a50009bf5678b0f23a2bb63abef1d6babd0592fcc44924f524e9654ff32c5
8cab03d133181078f1b7ed2ec17077131084156ff9ece96c0b92681451928607d63b71d38dbe6679d08f65
d4347cfd10c8d5232ff7b4b9555f9742b35e850eed308fe4b06669cf2096ac298b28a04d45c0f44e5fa3f88d
a1b4a188988d01c709b9483c531ad96d84cd4b4c46f3bcdbbdf37a76f830b5716984365c2787cf1a61163fa
328470ff03b03848746ad8925ae005521eddb082f15a2d9cf94092a438841446a3e3bc00c7c421e5c81b48
b5d8b37a1f1370b7064f9b8603677f1dd6426321a9ffec4590aad76f68b1dfd2c73094be9cafbe737e7d2344
24c06af151662e48af24569b9cb93771ab6227039a453534ad034d23d1efefad8487b43abe4b1e71a4d426
5e2a4b093406634a0e8e24bb1cc4024ada401b4f3660b271443981c914dbf7a06369a727da3faae84799d2
cc0fc5f16f110f98d9fb2531b9c2ae53c96689092919d8be80e8ddaaa5bad2dedaa4db7378bf5fef68af6df5ff
d5736b0284685dc56fd06bc1b25312e4a9d0a57d62da93be67656a35fb02d3abac1cdd4bf1c564d8139511
05843cf6c207f8f2b3a466ffecbb2cc3dc633b1aea231505b4b3a7314cf197bdd1388e94fc1d186ad44410fcf
1833949474cf62ca432a970bfecdfe9382409da64ac308dadfc300f7dcffe908023f30ccf35b0d185c4af4f25
362654feeaea9ed5616b2d8ebae7462c39f363be41b686c31f9e1377bac296365379d614d87dd487985
fdae96e6ca82a5d9afda89b5b58dd3d4f1c078a6e785f43043d1abea6030b140df986f4ef74fb9d7a77d30e3
4bdf14ec3998dbb8669c70dcefd49040dd6a6aa814bf01911a578d60956b2426e14245aa
beb6e05cd6cf237b8bf78b59e68436cd2cfffaae69f2b7b753972dc24233108e2f00c8ab47b87d4e8d68121
970759cedf721ba93f0d753e47da27f126af895d73481922d7b52e1765db06e78fdf4ffc935eedc36d3e27b3
89f9c87f77314a953a45aaa9cd62efccdaee5c10c8db6d063c80b8542a0588614700f7d2b1062515d68a733
9c0846e73fa4e832745b3d750c68553375e5fafeeca31b803e28b134178ac7bda7dfc2f1e171a6978c5dd6d
03aeb38eed9ec08ecb3559d62602fb04927f9b16c5a19ea81d430c05d1e57b4067afb63e47e27e9da105b2f
700643c297cb23c442be0e76b2a67c5d7492388915fb401efafec290ab2304dc75490844e2d98b34b393aff
b0dc8dd5d731fdeac759a5abbe894555d693010766a0485b6aada7c59a544bcd1d6e5539dd2954f8499cbb
591a851917f3c1aeeba233c6fc5277b31c42456e1879e23acdf0fef5fd928d

输出：

80010000000e0000000080000000

B.19.3 TCM2_FlushContext

输入：

80010000000e0000016502000000

输出：

80010000000a00000000

B.19.4 TCM2_EvictControl

输入：

8002000000630000012040000001810000000000004902000000010341ff734c2ae8394e698cb7ffcc31

05c010030de8bce697fbf0ca3909b00e5102773cd0be05e233d6208e7986f2e19a2ffa004f64b69f07c1a80b
ec505846cb9cade3d81000000

输出：

8002000000530000000000000000010bc72eb69955a2b79143e77d41157517d010030ebe8b44cea716f
6b3ffdb0330fa5defe3635172093fd8d07e78fc146d7b473bd9e83ed9aeb01415cd29c33b643607676

B.20 属性命令输入输出实例

B.20.1 TCM2_GetCapability

输入：

800100000016000001 7A00000001000000000000040

输出：

8001000000730000000000000000010000001800000000000000010000000200000000300000004000000
0050000000600000007000000080000000900000000A0000000B0000000C0000000D0000000E000000
0F000000100000001100000012000000130000001400000015000000160000 0017

B.20.2 TCM2_TestParms

输入：

80010000001a0000018a00010006008000430010040000000000

输出：

80010000000a00000000

B.21 NV操作命令输入输出实例

B.21.1 TCM2_NV_DefineSpace

输入：

8002000000510000012A4000000100000000940000000900000000000000047061737300 2E010000000012
020A0088002079CB537C7A8D15834AB518A2B1D0AB4D30897F8D2B4067CAC9203D55129DC
4D900

输出：

8002000000130000000000000000000000010000

B.21.2 TCM2_NV_UndefineSpace

输入：

80020000001F000001224000000101000101000000094000000900000000000

输出：

8002000000130000000000000000000000010000

B.21.3 TCM2_NV_ReadPublic

输入：

80010000000e0000016901000000

输出：

80010000005E00000000002E010000000012220A0098002079CB537C7A8D15834AB518A2B1D0A

B4D30897F8D2B4067CAC9203D55129DC4D9000800220012EA652FC466FBAF73E9F9EE2256
4BC0949178820F462EADB62B742B3546EC8A3B

B.21.4 TCM2_NV_Write

输入：

8002000000360000013701000000010000000000000C4000000900000000036E6E6E00103132333435
36373839303132333435360000

输出：

8002000000130000000000000000000000010000

B.21.5 TCM2_NV_Increment

输入：

80020000005f000001340100059501000595000000490200000040010f06ae9b94a2f1442519d0a36f0ef8
70c01003092ab9215b716f19692b153bfe92de37f6c03793e6b2ef4f5aec31e9ef4454f295fc9c29a6c97474
46523801bb666f69d

输出：

8002000000530000000000000000001052f93b0e9f284d6a1b4a2c30ab5e244701003053e253c5cae7ab5
0b086ba0365e6b453521415d0c282f48cfc3f803a2fbf54a409ed948613e3199fc391a41d80d2c67d

B.21.6 TCM2_NV_Extend

输入：

80020000016b000001364000000101000c080000004902000000001046939cfda41246953cb967e5f045
12800100308b92ab68c72a1a68b24f2fc60a748259f91d8404573c4527fc0a112f3b6786338b3ec919d182
da0a7dc6978546660b0d010ad543e122bd2a7e20c0091c907847727c2bdcd66526a7345eb64b0d2aa5a13
af8baf1682b3f1ab1849280d0cfd7d618d78621bfdb907d7e8376064944e69ac05b81368393af19b48b9cf1
c9df61e15f3b4773051f6d493c800fe2a31212767812ad4c828812ab8d124b5d8532749543c9874ccf6b68
0ed01b053f055986a9e3e78630e88c5dc00369dd6785470a13977c48135557e24fc4eed1ecd643aab93668
a977c881389b019e248567f4934ce34e0e85842c7de1b8bffcdca96f9458cd74c85fa398a56e67921375924
8702aaa0946ee463448872472d50bd109f5dbec2ecd3207098e0f7670d84c523adc2b0bb8f5960dfd8f9b5
29d2e9b5919edcdd4dfa80e846bbb36e0a3fd39

输出：

8002000000530000000000000000001040ab042287325edaf8144deb2dad71d60100302b9b1f5d325904
dd031679f196ca239958dea3554dedd4024d2941bc8524d5a2e90587e89b31386bbc304b18b3fb8579

B.21.7 TCM2_NV_SetBits

输入：

8002000000670000013501000f4801000f4800000049020000030010f3a96211673606f616939865f9efb8
7b010030f96f12e6378b26cedc6be1492c22aa3ad31df21fa4d6bb555354f80eed76d0652defa544fe46821
3bbc3dcb7818a33ccf44c919a51ec5b49

输出：

8002000000530000000000000000010d9a3e9e593564b2e69f9c8a1797964ef01003009189ab48132a72
34d054f29e3f873a03e751fe9279bf42ab916346b1cff5ea1c101693c715a51ca403f898658913e12

B.21.8 TCM2_NV_WriteLock

输入：

8002000000220000013801000000010000000000000c4000000900000000036e6e6e

输出：

8002000000130000000000000000000000010000

B.21.9 TCM2_NV_GlobalWriteLock

输入：

80020000001b000001324000000c0000000940000009000000000

输出：

8002000000130000000000000000000000010000

B.21.10 TCM2_NV_Read

输入：

8002000000630000014e0100026a0100026a00000049020000000001041c5cb12346db58373cddb627cd7ddb80100300615430a0f4f216fbc919eef80e8c052b3e9e4b5d370db29a8e0f504ec2544768e75d8f60dc170797686c5f765b49c2000020005

输出：

8002000000570000000000000000040002ca0e001089d18d582e9070642a5e0bd91d2ef6a0010030464456 68ee0ca2f1a6c4cad31267186c120bd3171e5a7edc852fbc06c6e7bb1f5277d6805734559612109 49f5fa60d2a

B.21.11 TCM2_NV_ReadLock

输入：

8002000000220000014f0100000001000000000000c4000000900000000036e6e6e

输出：

8002000000130000000000000000000000010000

B.21.12 TCM2_NV_ChangeAuth

输入：

80020000006e0000013b0008e6ef00000049030000000205cf05b63f28cfb792a54e57f407e98bba8ed930b82d884547348c6da301f9cfa010020049396c55722ffb2159454ce192176d67389422f5541da579390d2068f0630860011644456bf2c666a8a01227c44c07f43b84f

输出：80010000000a000000000

附 录 C
（资料性）
与参考标准章条编号对照情况

本文件部分内容参考了 ISO/IEC 11889-1～11889-3:2015，与参考标准间的章条编号对照情况如下：

a） 第 5 章中参考内容见表 C.1；
b） 第 7 章中参考内容见表 C.2；
c） 附录 A 中参考内容见表 C.3。

表 C.1 第 5 章和附录 C 与 ISO/IEC 11889-1:2015 章条编号对照情况

本文件章条编号	ISO/IEC 11889-1:2015 章条编号
第5章可信计算密码支撑平台5.2～5.6	9 Tursted Platforms

表 C.2 第 7 章与 ISO/IEC 11889-3:2015 章条编号对照情况

本文件章条编号	ISO/IEC 11889-3:2015 章条编号
7.2 启动命令	10 Start-up
7.3 检测命令	11 Testing
7.4 会话命令	12 Session Commands
7.5 对象命令	13 Object Commands
7.6 复制命令	14 Duplication Commands
7.7 非对称算法命令（增加 7.7.2 和 7.7.3，2015 版 ISO 准章节中无对应内容）	15 Asymmetric Primitives
7.8 对称算法命令	16 Symmetric Primitives
7.9 随机数发生器命令	17 Random Number Generator
7.10 杂凑/HMAC命令	18 Hash/HMAC/Event Sequences
7.11 证明命令	19 Attestation Commands
7.12 临时EC密钥命令	20 Ephemeral EC Keys
7.13 签名及签名验证命令	21 Signing and Signature Verification
7.14 度量命令	23 Integrity Collection(PCR)
7.15 增强授权命令	24 Enhanced Authorization(EA)Commands
7.16 分层命令	25 Hierarchy Commands
7.17 字典攻击命令	26 Dictionary Attack Functions
7.18 管理功能命令	27 Miscellaneous Management Functions
7.19 上下文管理命令	29 Context Management
7.20 属性命令	31 Capability Commands
7.21 NV操作命令	32 Non-volatile Storage

表 C.3　附录 A 与 ISO/IEC 11889-2:2015 章条编号对照情况

本文件章条编号	ISO/IEC 11889-2:2015 章条编号
A.1命令码	7.5TPM_CC(Command Codes)
A.2返回码	7.6TPM_RC(Response Codes)
A.3基本常量	5Base Types 6Constants 7Handles 9Interface Types
A.4结构定义	10Structure Definitions
A.5密码参数和结构	11Algorithm Parameters and Structures
A.6密钥/对象结构	12Key/Objects Complex
A.7NV存储结构	13NV Storage Structures
A.8上下文数据	14Context Data

附 录 D
（资料性）
典型应用中接口依赖关系示例

D.1 签名

签名接口的依赖关系见图 D.1。

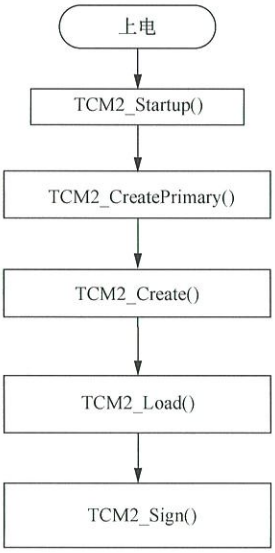

图 D.1 签名接口依赖关系

D.2 验证签名

验证签名接口的依赖关系见图 D.2 和图 D.3。

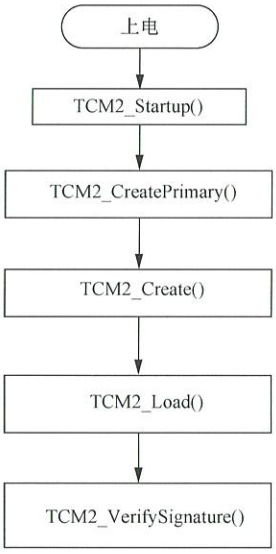

图 D.2 芯片内部的密钥完成签名验证接口依赖关系

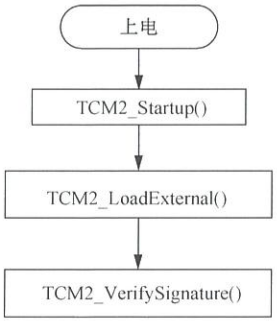

图 D.3　由外部导入密钥完成签名验证口依赖关系

D.3　非对称加密

非对称加密接口的依赖关系见图 D.4 和图 D.5。

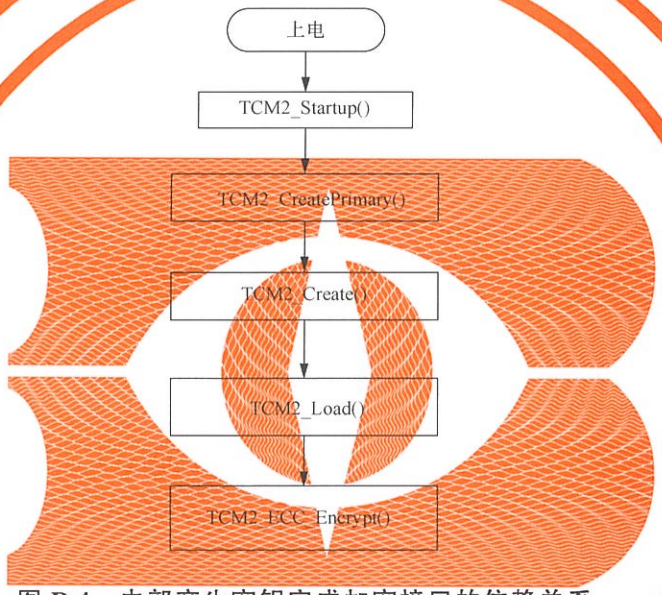

图 D.4　内部产生密钥完成加密接口的依赖关系

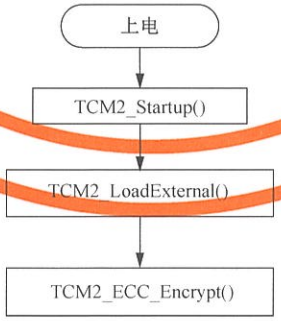

图 D.5　外部导入密钥完成加密接口的依赖关系

D.4　非对称解密

非对称解密接口的依赖关系见图 D.6。

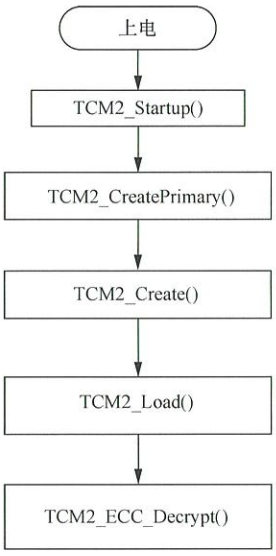

图 D.6　非对称解密接口的依赖关系

D.5　杂凑

杂凑接口的依赖关系见图 D.7。

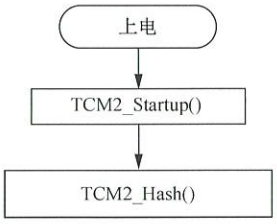

图 D.7　杂凑功能接口的依赖关系

D.6　对称加密

TCM2_EncryptDecrypt()中 TCMI_YES_NO 为 NO 时为加密，对称加密接口的依赖关系见图 D.8 和图 D.9。

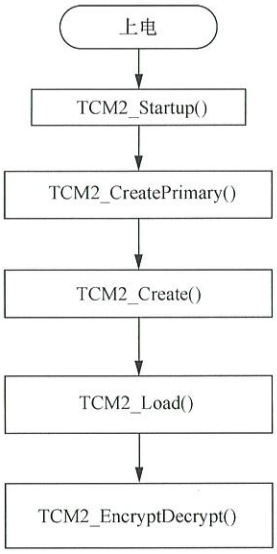

图 D.8　内部产生对称密钥完成对称算法加密功能接口的依赖关系

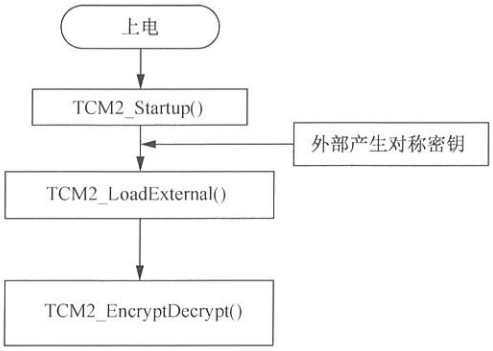

图 D.9　外部产生对称密钥完成对称算法加密功能接口的依赖关系

D.7　对称解密

TCM2_EncryptDecrypt()中 TCMI_YES_NO 为 YES 时为解密，对称解密接口的依赖关系见图 D.10 和图 D.11。

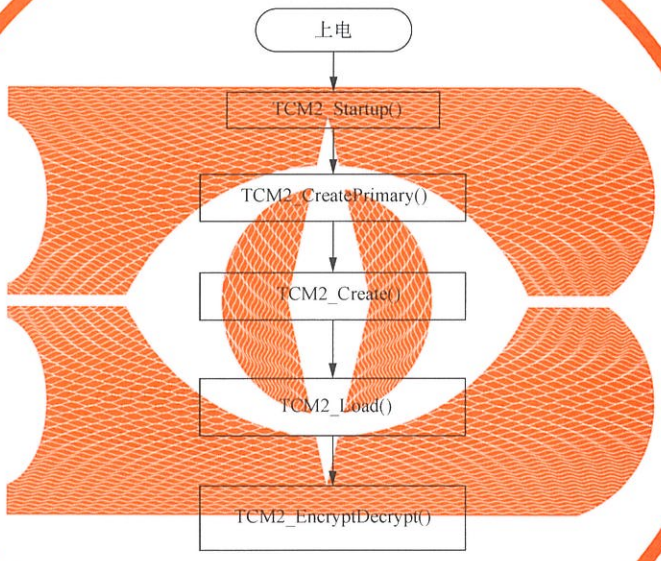

图 D.10　内部产生对称密钥完对称算法解密密功能接口的依赖关系

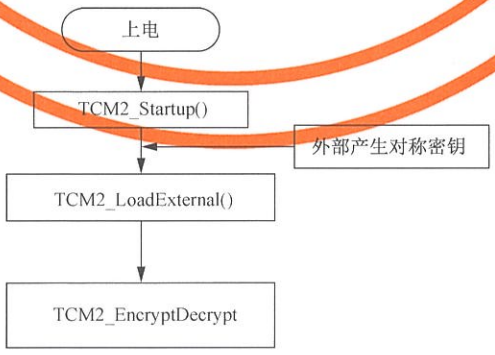

图 D.11　外部产生对称密钥完成对称算法加密功能接口的依赖关系

D.8　远程证明

远程证明的接口依赖关系见图 D.12。

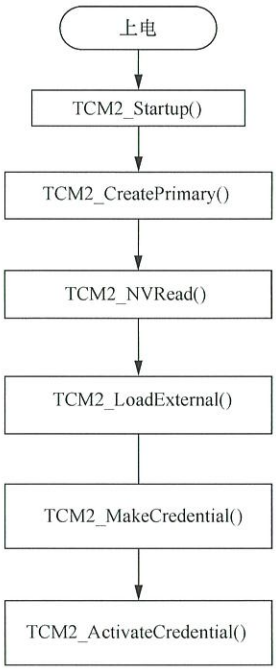

图 D.12　远程证明应用中的功能接口的依赖关系

参 考 文 献

[1] GB/T 18336(所有部分) 信息技术 安全技术 信息技术安全评估准则

[2] GB/T 22239—2019 信息安全技术 网络安全等级保护基本要求

[3] GB/T 29827—2013 信息安全技术 可信计算规范 可信平台主板功能接口

[4] GB/T 29828—2013 信息安全技术 可信计算规范 可信连接架构

[5] GB/T 30847.1—2014 系统与软件工程 可信计算平台可信性度量 第1部分:概述与词汇

[6] GB/T 30847.2—2014 系统与软件工程 可信计算平台可信性度量 第2部分:信任链

[7] GB/T 36639—2018 信息安全技术 可信计算规范 服务器可信支撑平台

[8] GB/T 37935—2019 信息安全技术 可信计算规范 可信软件基

[9] GB/T 38638—2020 信息安全技术 可信计算 可信计算体系结构

[10] GB/T 38644—2020 信息安全技术 可信计算 可信连接测试方法

[11] ISO/IEC 27070 Information technology—Security techniques—Security requirements for virtualized roots of trust, 2nd WD

[12] ISO/IEC 27071: 2023 Cybersecurity—Security recommendations for establishing trusted connections between devices and services

[13] ISO/IEC 11889-1: 2015 Information technology—Trusted Platform Module Library—Part 1: Architecture

[14] ISO/IEC 11889-2: 2015 Information technology—Trusted Platform Module Library—Part 2: Structure

[15] ISO/IEC 11889-3: 2015 Information technology—Trusted Platform Module Library—Part 3: Commands