ICS 35.040
CCS L 80

GM

# 中华人民共和国密码行业标准

GM/T 0012—2020
代替 GM/T 0012—2012

# 可信计算　可信密码模块接口规范

**Trusted computing—Trusted computing interface specification of trusted cryptography module**

2020-12-28 发布　　2021-07-01 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 0012—2012《可信计算　可信密码模块接口规范》。

本文件与GM/T 0012—2012相比，主要的技术变化如下：

a) 修改了前言和引言的内容；

b) 修改了原有标准第3章“术语、定义和缩略语”，按照GB/T 1.1—2020的要求修改为第3章，并修改和增加了术语和定义的内容；

c) 删除原有标准的第4章“概述”内容；

d) 增加了第4章“缩略语”，且增加和修改了部分内容；

e) 删除原有标准的第5章、第6章、第7章、第8章内容；

f) 增加了第5章“可信密码模块功能概述”内容；

g) 增加了第6章“可信密码模块功能接口”内容，该内容参照了ISO/IEC 11889-3：2015；

h) 增加了6.7，关于SM2非对称加解密的指令的实现要求；

i) 修改了规范性附录A“数据结构”，该内容参照了ISO/IEC 11889-2：2015。

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、联想控股有限公司、山谷网安科技股份有限公司、同方股份有限公司、中国科学院软件所、长春吉大正元信息技术股份有限公司、中国长城计算机深圳股份有限公司、成都卫士通信息产业股份有限公司、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、北京信息科技大学、北京卓识网安技术股份有限公司、北京天融信网络安全技术有限公司。

本文件主要起草人：范琴、刘鑫、付月朋、秦宇、谷晶中、吴秋新、杨贤伟、邹浩、余发江、宁晓魁、王梓、郑必可、林洋、李伟平、雷晓锋、徐震、姚金龙、严飞、李丰、许勇、贾兵、王蕾、顾健、何长龙、刘韧。

本文件所代替文件的历次版本发布情况为：

——GM/T 0012—2012。

# 引　言

本文件描述了可信计算　可信密码模块接口规范。通过本文件的接口，向应用层提供统一的 TCM 接口，适用于可信计算应用的开发、使用及检测并提供标准依据和指导，有利于提高可信计算产业发展水平。

本文件在 GM/T 0012—2012《可信计算　可信密码模块接口规范》的基础上，参考了 TPM2.0 标准(ISO/IEC 11889:2015)相关内容进行了修订。

本文件支持中国密码 SM2、SM3、SM4 算法，在功能上与 TPM2.0 标准中使用中国算法模式的内容兼容。在密码算法使用设计上，未来密码算法如密钥长度升级，杂凑算法摘要长度升级等，相关接口的设计上兼容后续算法的更新或新增密码算法。预留国际算法模式，为中国可信计算走向国际奠定基础。

《可信计算　可信密码模块接口规范》为芯片接口规范，可信计算密码支撑平台功能与接口规范是为应用层提供服务的接口规范，是对可信密码模块接口规范的接口进行的封装，厂商或者开发者可以在可信计算密码支撑平台功能与接口规范的产品中开发兼容可信密码模块接口规范中定义的接口。

可信计算密码支撑平台功能原理相关内容，请参考 GB/T 29829 相关章节。

# 可信计算　可信密码模块接口规范

## 1　范围

本文件描述了可信密码模块的功能，详细定义了可信密码模块的命令接口。

本文件适用于可信密码模块相关产品的研制、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　数字证书格式规范

GB/T 29829　信息安全技术　可信计算密码支撑平台功能与接口规范

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（所有部分）　信息安全技术　SM2 椭圆曲线公钥密码算法

## 3　术语和定义

下列术语和定义适用于本文件。

3.1

**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。

[GB/T 29829—2013，3.1.1]

3.2

**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、密码服务等内容，为可信计算平台自身的完整性、身份可信性和数据安全性提供密码支持。其产品形态主要表现为可信密码模块和可信密码服务模块。

[GB/T 29829—2013，3.1.2]

3.3

**完整性度量　integrity measurement**

使用密码杂凑算法对被度量对象计算其杂凑值的过程。

[GB/T 29829—2013，3.1.3]

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。

[GB/T 29829—2013，3.1.4]

3.5

**可信存储根　root of trust for storage**

指存储主密钥，是可信计算平台内进行可信存储的基础。

[GB/T 29829—2013，3.1.5]

3.6

**可信报告根　root of trust for reporting**

指密码模块密钥，是可信计算平台内进行可信报告的基础。

[GB/T 29829—2013，3.1.6]

3.7

**可信密码模块　trusted cryptography module**

是可信计算平台的硬件模块，为可信计算平台提供密码运算功能，具有受保护的存储空间。

[GB/T 29829—2013，3.1.7]

3.8

**TCM 服务模块　TCM service module**

可信计算密码支撑平台内部的软件模块，为对平台外部提供访问可信密码模块的软件接口。

[GB/T 29829—2013，3.1.8]

3.9

**部件　Component**

计算系统中可被度量的硬件和/或软件模块。

[GB/T 29829—2013，3.1.9]

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。

[GB/T 29829—2013，3.1.10]

3.11

**完整性度量值　integrity measurement value**

部件被度量后得到的杂凑值。

[GB/T 29829—2013，3.1.11]

3.12

**完整性基准值　predefined integrity value**

部件在可信状态下被度量得到的杂凑值。该值可作为完整性校验的基准。

[GB/T 29829—2013，3.1.12]

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部件之间所建立的信任传递关系。

[GB/T 29829—2013，3.1.13]

3.14

**可信方　trusted party**

提供可信证明的机构，包含可信第三方和主管方。

[GB/T 29829—2013，3.1.14]

3.15

**密钥管理中心　key management centre**

用于密钥生成和管理的软硬件系统。

[GB/T 29829—2013，3.1.15]

3.16

**双证书　dual certificate**

包括签名证书和加密证书，分别用于签名和数据加密，由可信方一同签发。

[GB/T 29829—2013，3.1.16]

3.17

**实体　entity**

访问密码支撑平台资源的应用程序和用户。

[GB/T 29829—2013，3.1.17]

3.18

**对象　object**

可信计算密码支撑平台内可以被实体访问的各类资源，包括密钥数据、运行环境数据、敏感数据等。

[GB/T 29829—2013，3.1.18]

3.19

**密码模块密钥　tcm endorsement key**

可信密码模块的背书密钥。

[GB/T 29829—2013，3.1.23]

3.20

**存储主密钥　storage master key**

用于保护平台身份密钥和用户密钥的主密钥。

[GB/T 29829—2013，3.1.26]

3.21

**直接匿名证明　direct anonymous attestation**

可信计算平台所使用的匿名身份鉴别方案。

3.22

**基于椭圆曲线的直接匿名证明　elliptic curve-based direct anonymous attestation**

基于椭圆曲线密码学方案的直接匿名证明方案。

3.23

**证明方　prover**

直接匿名证明中的证明方，一般包含主机和可信密码模块两个部分。

3.24

**安全主机　security host**

直接匿名证明中证明方拥有的内嵌可信密码模块的主机。

3.25

**凭证颁发方　Issuer**

直接匿名证明中，为可信密码模块颁发凭证的参与方。

3.26

**验证方　verifier**

直接匿名证明中，验证远程可信密码模块身份的参与方。

## 4　缩略语

下列缩略语适用于本文件。

DAA　直接匿名证明(Direct Anonymous Attestation)

ECDAA　基于椭圆曲线的直接匿名证明(Elliptic Curve-based Direct Anonymous Attestation)
ECDH　基于椭圆曲线的密钥交换(Elliptic Curve Diffie-Hellman)
EK　初始密钥(Endorsement Key)
EPS　初始密钥主种子(Endorsement Primary Seed)
HMAC　带密钥的消息验证码算法(the keyed-Hash Message Authentication Code)
KDF　密钥导出函数(Key Derivation Function)
KMC　密钥管理中心(Key Management Center)
NV　非易失性(Non-Volatility)
PCR　平台配置寄存器(Platform Configuration Register)
PPS　平台主种子(Platform Primary Seed)
SPS　存储主种子(Storage Primary Seed)
TCM　可信密码模块(Trusted Cryptography Module)
TSM　TCM 服务模块(TCM Service Module)

## 5　可信密码模块功能概述

### 5.1　可信计算平台

可信计算平台主要由 TCM 和 TSM 两大部分组成,其功能架构见图 1。

注:图 1 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。

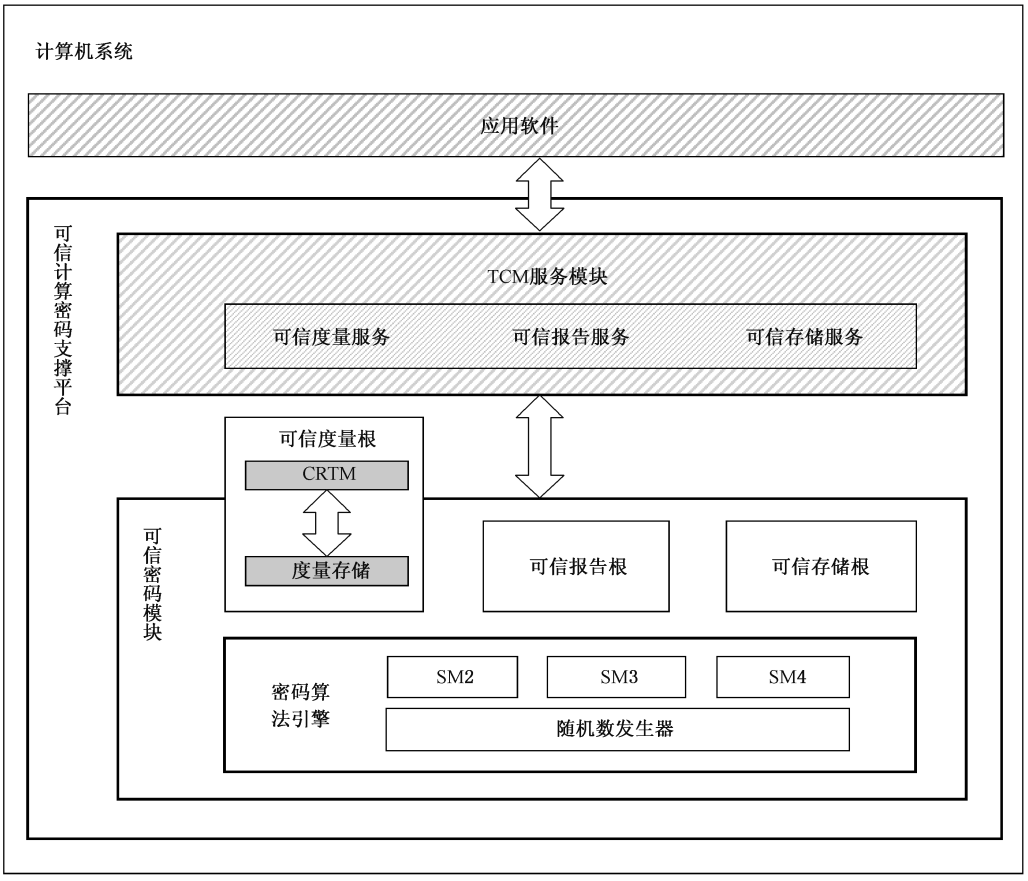



图 1　可信计算平台功能架构

可信计算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性，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。

可信计算平台利用可信密码模块内的密码机制，通过对系统平台组件的完整性度量、存储与报告，确保平台完整性。

完整性度量与存储是指计算部件的度量值，记录该事件到事件日志，并把度量值记入可信密码模块内相应的平台配置寄存器(PCR)中。

可信计算密码支撑平台功能原理相关内容按照 GB/T 29829 相关章节执行。

图 2 为依次度量两个部件的过程。

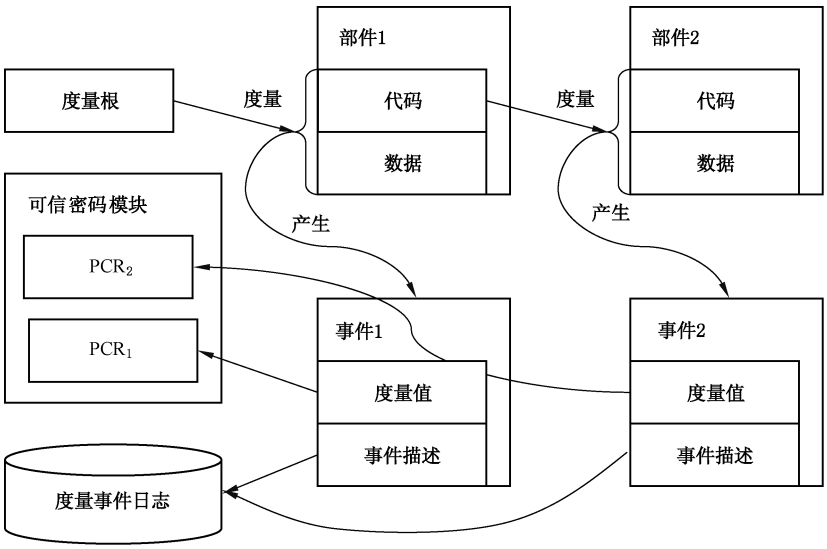



**图 2　完整性度量流程**

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以表征被度量者特性的数据；

c) 杂凑运算输出的杂凑值即为被度量者的完整性度量值；

d) 度量者应把度量值记入指定的 PCR 中。记入的办法是：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：度量者信息、被度量者信息、原 PCR 值、度量值、新 PCR 值、完成时间等；

f) 如果一个部件序列中的各部件完整性度量值存储在同一个 PCR 中，则采用一种专门的压缩存储方式，即从第一个部件开始，将该部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，最后一个部件的完整性度量值存储操作完成后，所得值即为该部件序列存储到 PCR 中的完整性度量值。

完整性报告是指平台向验证者提供平台或部分部件的完整性度量值的过程。

完整性报告应满足如下要求：

a) 平台能够向验证者提供指定 PCR 值，无需任何授权；

b） 平台能够向验证者提供指定 PCR 值以及对 PCR 值的签名。签名使用平台身份密钥；

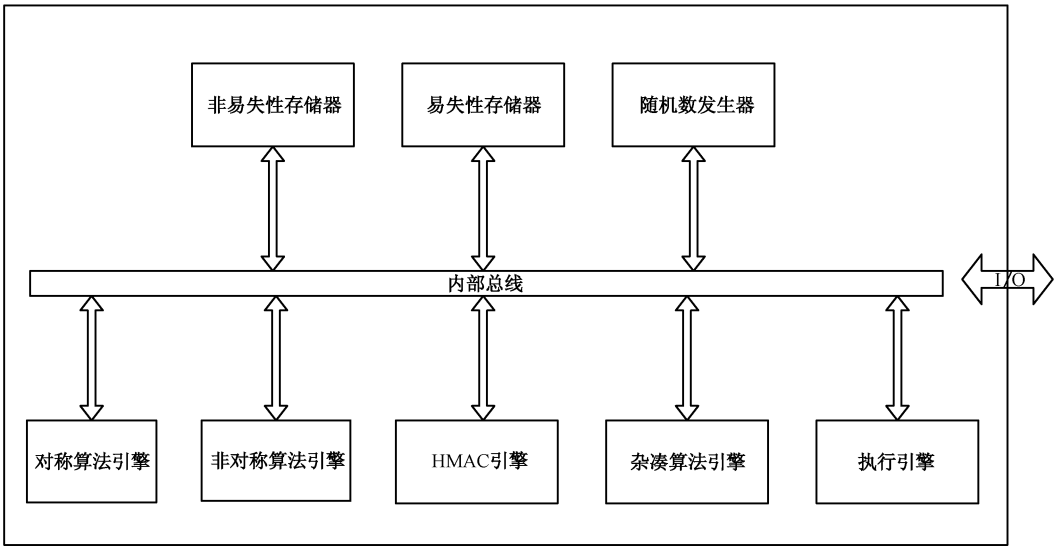
c） 平台可向验证者提供指定 PCR 的相关事件日志信息；

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。

### 5.2 可信密码模块

TCM 是可信计算密码支撑平台必备的关键基础部件，提供独立的密码算法支撑。TCM 是硬件和固件的集合，可以采用独立的封装形式，也可以采用 IP 核的方式和其他类型芯片集成在一起，提供 TCM 功能。其结构见图 3。




各部件说明：

非易失性存储器：存储永久数据的存储单元；

易失性存储器：TCM 运行时临时数据的存储单元；

随机数发生器：生成随机数的单元；

内部总线：可信密码模块内部数据传输的总线；

I/O：TCM 的输入输出硬件接口；

对称算法引擎：执行对称密码运算的单元；

非对称算法引擎：产生非对称密钥对和执行非对称加/解密、签名运算的单元；

HMAC 引擎：基于杂凑引擎的计算消息认证码单元；

杂凑算法引擎：执行杂凑运算的单元；

执行引擎：TCM 的运算执行单元。

**图 3 可信密码模块结构**

## 6 可信密码模块功能接口

### 6.1 通用要求

本章节规定了可信密码模块功能接口的具体命令。本文件所定义的可信密码模块应满足本章节所有的命令接口。

## 6.2 启动命令

### 6.2.1 TCM2_Startup

该命令用于 TCM 初始化，当该命令执行成功后，不再允许成功执行该命令。

当 TCM 需要 TCM2_Startup 命令时，收到其他命令，或者当不需要此命令时收到了该命令，TCM 将返回 TCM2_RC_INITIALIZE。

如下是关闭/启动序列，定义了 TCM 收到 TCM2_Startup()命令后的操作方法。

a) TCM 重置：发送 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。

b) TCM 重启：发送 Shutdown(STATE)命令关闭，启动时发送 Startup(CLEAR)。此状态下的如下值会恢复到初始状态：
   - 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 Shutdown(STATE)命令关闭，启动时发送 Startup(STATE)。此状态下的如下状态将会被保存：
   - S-RTM；
   - PCR；
   - 除 phEnable、phEnableNV 外的平台控制开关。

本文件的所有接口涉及的命令码、返回码及常用和数据结构，请见附录 A。

该命令接口输入和输出参数及说明见表 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 |

### 6.2.2 TCM2_Shutdown

该命令用于为一个新的电源周期准备 TCM。

shutdownType 参数用于指明后续的 TCM2_Startup 将采用怎么的处理过程。

该命令接口输入和输出参数及说明见表 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 |

## 6.3　检测命令

### 6.3.1　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 将只对为执行测试的能力进行自检。

如果需要自检，TCM 将：

返回 TCM2_RC_TESTING，执行需要进行的功能自检；

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及说明见表 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：执行全测试<br>NO：对未执行测试功能执行测试 |

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 |

### 6.3.2 TCM2_IncrementalSelfTest

该命令对被选中的算法执行测试。

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

该命令接口输入和输出参数及说明见表 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 | 将被测试的算法列表 |

**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 |
| TCM2_RC | responseCode | 返回码 |
| TCML_ALG | toDoList | 需要被测试算法列表 |

### 6.3.3 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参数及说明见表 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_GetTestRusult |

**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 | 将被自检的算法列表 |

## 6.4 会话命令

### 6.4.1 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非对称保护方式往 TCM 注入秘密信息。TcmKey 的类型决定如何加密 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。

该命令接口输入和输出参数及说明见表 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<br>可以是 TCM2_RH_NULL<br>授权索引：None |
| TCMI_DH_ENTITY+ | bind | 实体提供授权值<br>可以是 TCM2_RH_NULL<br>授权索引：None |
| TCM2B_NONCE | nonceCaller | 初始的暂时调用，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<br>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(包括试验策略) |
| TCMT_SYM_DEF+ | symmetric | 参数加密的算法及密钥长度，可以为 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，则输出标记应该是 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 |

### 6.4.2 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参数及说明见表 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 |

## 6.5 对象命令

### 6.5.1 TCM2_Create

该命令用于创建一个可以被 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。该对象使用之前需要调用 TCM2_Load 命令载入 TCM。

该命令接口输入和输出参数及说明见表 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，父句柄应该指向一个已载入 TCM 的解密密钥<br>授权索引：1<br>授权角色：USER |
| TCM2B_SENSITIVE_CREATE | inSensitive | 隐私数据 |
| TCM2B_PUBLIC | inPublic | 公开数据 |
| TCM2B_DATA | outsideInfo | 将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以认证的联系 |
| 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,则输出标记应是 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 进行计算的摘要值 |
| 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) |

### 6.5.2　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)都需要被载入 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。

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参数及说明见表 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<br>授权索引:1<br>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，则输出标记应是 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 |

### 6.5.3　TCM2_LoadExternal

此命令用于将一个非保护性的对象载入 TCM，该命令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参数及说明见表 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 | TCM2_ST_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，则输出标记应是 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 |

### 6.5.4 TCM2_ReadPublic

该命令返回一个已载入对象的公开数据部分，该对象句柄的使用不需要授权信息。

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参数及说明见表 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，则是 TCM2_ST_SESSIONS，否则应是 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<br>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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_PUBLIC | outPublic | 该对象的公开数据部分 |
| TCM2B_NAME | name | 对象的名字 |
| TCM2B_NAME | qualifiedName | 对象的限定名称 |

### 6.5.5 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在 KDF 中用于恢复对称密钥。恢复的种子（而不是名称）在 KDF 中用于恢复 HMAC 密钥。

HMAC 被用来认证输入参数 credentialBlob 是否与 activateHandle 相关联，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参数及说明见表 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 | credentialBlob 中与证书关联的对象的句柄<br>授权索引:1<br>授权身份:ADMIN |
| TCMI_DH_OBJECT | @keyHandle | 已经加载的用来解密 credentialBlob 中隐私数据的密钥<br>授权索引:2<br>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,则输出标记应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_DIGEST | certInfo | 解密的证书信息<br>数据不应大于与 keyHandle 关联的 nameAlg 摘要的大小 |

### 6.5.6　TCM2_MakeCredential

该此命令允许 TCM 执行证书颁发机构(CA)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参数及说明见表 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,则是 TCM2_ST_SESSIONS,否则应为 TCM2_ST_NO_SESSIONS |
| UINT32 | commandSize | 命令长度 |

表 25　TCM2_MakeCredential()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2_CC | commandCode | TCM2_CC_MakeCredential |
| TCMI_DH_OBJECT | handle | 载入公开区域的句柄<br>用来加密包含凭证密钥的敏感数据<br>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,则输出标记应是 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 |

### 6.5.7　TCM2_Unseal

该命令返回一个被载入 TCM 的封装数据对象的相关数据。

这个封装数据对象可是 TCM 自己产生的,但更有可能是外部创建,通过 TCM2_Import 导入 TCM 的。该命令的返回值通过授权会话加密。

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

该命令接口输入和输出参数及说明见表 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<br>授权索引:1<br>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，则输出标记应是 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<br>解封数据的大小不能超过 128 字节 |

### 6.5.8　TCM2_ObjectChangeAuth

此命令用于更改 TCM 驻留对象的授权机密。如果命令执行成功，则返回与 objectHandle 关联的 TCM 驻留对象的新专用区域，其中包括新的授权值。此命令不会更改其操作所在的 TCM 常驻对象的授权。因此，如果需要，在生成响应 HMAC 密钥时使用(TCM 驻留对象的)旧 authValue。

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需要调用 TCM2_CreatePrimary 命令重新创建该对象。

该命令接口输入和输出参数及说明见表 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<br>授权索引：1<br>授权身份：ADMIN |
| TCMI_DH_OBJECT | parentHandle | 父密钥句柄<br>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，则输出标记应是 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 |

## 6.6 复制命令

### 6.6.1 TCM2_Duplicate

该命令复制一个已加载的对象使得该对象可以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。

此命令需要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参数及说明见表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<br>授权索引:1<br>授权身份:DUP |
| TCMI_DH_OBJECT+ | newParentHandle | 引用非对称密钥的公共区域<br>授权索引:None |
| TCM2B_DATA | encryptionKeyIn | 可选的对称加密密钥<br>如果该密钥由TCM生成,密钥长度被设置为0<br>该参数可以是被加密的 |
| TCMT_SYM_DEF_OBJECT+ | symmetricAlg | 内部加密使用的对称加密算法<br>如果不存在内部加密,则该参数应该是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,则输出标记应是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 | 可以用encryptionKeyIn加密并可以双重加密的私有区域 |
| TCM2B_ENCRYPTED_SECRET | outSymSeed | 新的父密钥的种子,由非对称算法保护 |

### 6.6.2 TCM2_Rewrap

该命令允许 TCM 充当复制权限的角色。

如果使用了提供的 oldParent 的正确授权，则从 inSymSeed 恢复 HMAC 密钥和对称密钥，并用于完整性检查和对 inDuplicate 进行解密。根据适用于 newParent 的方法生成新的保护种子值，并对 blob 重新加密并计算新的完整性值。重新加密的 blob 在 outDuplicate 中返回，对称密钥在 outSymKey 中返回。

该命令接口输入和输出参数及说明见表 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<br>授权索引：1<br>授权身份：User |
| TCMI_DH_OBJECT+ | newParent | 对象的新父密钥句柄<br>授权索引：None |
| TCM2B_PRIVATE | inDuplicate | 使用从 inSymSeed 派生的对称密钥加密的对象 |
| TCM2B_NAME | name | 被迁移的对象名字 |
| TCM2B_ENCRYPTED_SECRET | inSymSeed | 对称密钥的种子<br>需要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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_PRIVATE | outDuplicate | 使用从 outSymSeed 派生的对称密钥加密的对象 |
| TCM2B_ENCRYPTED_SECRET | outSymSeed | 由 newParent 非对称密钥保护的对称密钥的种子 |

### 6.6.3 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，并在一个新的层级上使用。被导入的对象可被一次或多次加密，或者没有加密。

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参数及说明见表 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 | 新对象的父密钥句柄<br>授权索引：1<br>授权身份：USER |
| TCM2B_DATA | encryptionKey | 用作复制的内部数据包的可选对称加密密钥<br>如果 symmetricAlg 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<br>此参数是为了检查 duplicate 和 object 属性的完整性值。<br>**注**：即使完整性值在输入的时候没有被检查，对象的名字被要求包含被导入的对象的完整性值 |
| 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<br>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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_PRIVATE | outPrivate | 被父密钥的对称密钥加密后的隐私部分 |

## 6.7 非对称算法命令

### 6.7.1 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,该命令需要与TCM2_EC_Ephemal()命令配合使用。

TCM2_EC_Ephemeral()命令用于生成短暂生命周期的密钥(简称短暂密钥)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(de,V)以及对应的公钥(Qe,V),keyA提供静态短暂元素ds,V和Qs,V.TCM将计算出Zs和Ze的值。

该命令接口输入和输出参数及说明见表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<br>该句柄的私钥为dS,A<br>授权索引:1<br>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,则输出标记应是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$X$和$Y$轴坐标 |
| TCM2B_ECC_POINT | outZ2 | 第二次计算值得$X$和$Y$轴坐标 |

### 6.7.2 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。

SM2 算法遵循 GB/T 32918(所有部分)。

该命令接口输入和输出参数及说明见表 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 |
| TCMT_ECC_DECRYPT | inScheme | Ecc 加密策略 |
| TCM2B_MAX_BUFFER | message | 待加密数据 |
| TCM2B_DATA | sharedData1 | (可选)如果加密策略是 TCMs_scheme_ecaes,该参数是强制的 |
| TCM2B_DATA | sharedData2 | (可选) 如果加密策略是 TCMs_scheme_ecaes,该参数是强制的 |

**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 应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_MAX_BUFFER | ciphexText | 已加密数据的输出 |

### 6.7.3　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。

SM2 算法遵循 GB/T 32918(所有部分)。

该命令接口输入和输出参数及说明见表 41 和表 42。

**表 41　TCM2_ECC_Dencrypt()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<br>授权索引:1 |
| TCMT_ECC_DECRYPT | inScheme | ECC 加密策略 |

表 41　TCM2_ECC_Dencrypt()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2B_MAX_BUFFER | ciphexText | 需解密的数据 |
| TCM2B_DATA | sharedData1 | (可选)如果解密策略是 TCMs_scheme_ecaes,该参数是强制的 |
| TCM2B_DATA | sharedData2 | (可选) 如果解密策略是 TCMs_scheme_ecaes,该参数是强制的 |

表 42　TCM2_ECC_Dencrypt()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_MAX_BUFFER | message | 已解密的数据输出 |

### 6.7.4　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(TCM_RC_KEY)。

被 keyHandle 引用的密钥方案必须是 TCM_ALG_ECDH 或 TCM_ALG_NULL(TCM_RC_scheme)。

inPoint 必须位于 keyHandle 引用的密钥的曲线上(TCM_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参数及说明见表 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<br>授权索引:1<br>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，则输出标记应是 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<br>$Z=(x_Z,y_Z):=[hd_s]Q_B$ |

### 6.7.5　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 不需要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 的返回值可以使用参数加密进行加密。

该命令接口输入和输出参数及说明见表 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，则是 TCM2_ST_SESSIONS，否则为 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 密钥的公共区域<br>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，则输出标记应是 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 | 产生临时的公共的点($Q_e$) |

## 6.8 对称算法命令

TCM2_EncryptDecrypt2 命令用于执行对称加解密操作。

keyHandle 参数执向对称密钥对象(TCM2_RC_KEY)。

对于受限密钥,mode 参数需要与密钥的模式相同,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)。

SM4 对称密码算法遵循 GB/T 32907。

该命令接口输入和输出参数及说明见表 47 和表 48。

**表 47 TCM2_EncryptDencrypt2()接口输入参数**

| 类型 | 名称 | 描述 |
|---|---|---|
| TCMI_ST_COMMAND_TAG | tag | TCM_ST_SESSIONS |
| UINT32 | commandSize | 命令长度 |
| TCM2_CC | commandCode | TCM_CC_EncryptDecrypt2 |
| TCMI_DH_OBJECT | @keyHandle | 用于操作的密钥句柄<br>授权索引:1<br>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<br>如果是 NO,当前为加密操作 |
| TCMI_ALG_CIPHER_MODE+ | mode | 对称模式<br>该字段应与密钥的默认模式匹配或为 TCM_ALG_NULL |
| TCM2B_IV | ivIn | 算法要求的初始值 |

**表 48 TCM2_EncryptDencrypt2()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_MAX_BUFFER | outData | 加密后或者解密后的数据 |
| TCM2B_IV | ivOut | 下一循环需要使用的 IV 值 |

## 6.9 随机数发生器命令

### 6.9.1 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。

产生的随机数满足 GB/T 32915 要求。

该命令接口输入和输出参数及说明见表 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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |
| TCM2B_DIGEST | randomBytes | 随机数 |

### 6.9.2　TCM2_StirRandom

该命令用于往随机数发生器中添加“附加信息”。

注：inData 参数的长度不能大于 128 字节。

该命令接口输入和输出参数及说明见表 51 和表 52。

**表 51　TCM2_StirRandom()接口输入参数**

| 类型 | 名称 | 描述 |
|---|---|---|
| TCMI_ST_COMMAND_TAG | tag | 如果是解密会话，则是 TCM2_ST_SESSIONS，否则为 TCM2_ST_NO_SESSIONS |
| UINT32 | commandSize | 命令长度 |
| TCM2_CC | commandCode | TCM2_CC_StirRandom {NV} |
| TCM2B_SENSITIVE_DATA | inData | 附加信息 |

表 52　TCM2_StirRandom()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 |

## 6.10　HASH/HMAC 命令

### 6.10.1　TCM2_Hash

该命令基于数据缓存区执行 SM3 杂凑运算并返回运算结果。

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，需要使用杂凑序列命令。

当杂凑运算的结果需要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能够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，票据中的和摘要 digest 设置为空。

SM3 杂凑算法遵循 GB/T 32905。

该命令接口输入和输出参数及说明见表 53 和表 54。

表 53　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 | 待杂凑运算的数据 |
| 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 54　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 |

表 54　TCM2_Hash()接口输出参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2B_DIGEST | outHash | 杂凑运算结果 |
| TCMT_TK_HASHCHECK | validation | 如果摘要没有使用受限密钥签名，则表明用于计算 outDigest 的八位字节序列不是以 TCM_GENERATED_VALUE 开头的票证是空票证 |

### 6.10.2　TCM2_HMAC

该命令使用提供的数据集指明的杂凑算法(SM3)执行 HMAC 运算。

执行者使用 handle 时需要提供恰当的授权信息。

当 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参数及说明见表 55 和表 56。

表 55　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 | 提供 HMAC 密钥的用于签名的密钥句柄<br>授权索引：1<br>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 56　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，则输出标记应是 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 |

### 6.10.3　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 时需要提供恰当的授权值。

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参数及说明见表 57 和表 58。

**表 57　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<br>授权索引：1<br>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 58　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，则输出标记应是 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 |

### 6.10.4　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。例外，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参数及说明见表 59 和表 60。

**表 59　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 |

表 59　TCM2_HashSequenceStart()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2B_AUTH | auth | 子序列使用该序列的授权值 |
| TCMI_ALG_HASH+ | hashAlg | 杂凑序列使用的杂凑算法<br>如果是 TCM_ALG_NULL，则为事件序列 |

表 60　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，则输出标记应是 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 |

### 6.10.5　TCM2_SequenceUpdate

该命令被用于给杂凑或者 HMAC 序列增加数据。

Buffer 中的数据可为小于 TCM 允许值的任何值。

sequenceHandle 参数相关的序列对象的使用需要恰当的授权。

如果与该命令关联的授权需要计算 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参数及说明见表 61 和表 62。

表 61　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<br>授权索引：1<br>授权角色：USER |
| TCM2B_MAX_BUFFER | buffer | 杂凑计算的数据 |

表 62　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.10.6 TCM2_SequenceComplete

该命令增加散列/HMAC 序列需要数据的最后部分，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 参数相关的序列对象的使用需要恰当的授权。如果与该命令关联的授权需要计算 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参数及说明见表 63 和表 64。

**表 63 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<br>授权角色：USER |
| TCM2B_MAX_BUFFER | buffer | hash/HMAC 运算的数据 |
| TCMI_RH_HIERARCHY+ | hierarchy | 杂凑操作的分层票据 |

**表 64 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，则输出标记应是 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 | 票据表明用于 outDigest 计算的数据序列不是以 TCM2_GENERATED_VALUE 开头<br>当为 HMAC 会话时，该值为 NULL |

## 6.11 证书命令

### 6.11.1 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。

objectHandle 的授权需要 ADMIN 角色的授权值。如果使用了增强授权会话,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 可以为 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参数及说明见表 65 和表 66。

本文件中证书格式遵循 GB/T 20518。

**表 65 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<br>授权索引:1<br>授权角色:ADMIN |
| TCMI_DH_OBJECT+ | @signHandle | 用于签名证明结构的密钥句柄<br>授权索引:2<br>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 66 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 应该是 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 |

### 6.11.2 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。并且该票证将验证加载的公共区域和提供的 creation 的数据的杂凑值(creationHash)之间的联系。

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 可以为 TCM2_Load()或者 TCM2_CreatePrimary()加载的任意对象。

该命令接口输入和输出参数及说明见表 67 和表 68。

**表 67　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 | 用于签证明数据块的密钥句柄<br>授权索引:1<br>授权角色:USER |
| TCMI_DH_OBJECT | objectHandle | 与产生的数据关联的对象<br>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 |

**表 68　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 应该是 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 |

### 6.11.3　TCM2_Quote

该命令用于引用 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参数及说明见表 69 和表 70。

**表 69　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<br>授权索引:1<br>授权角色:USER |
| TCM2B_DATA | qualifyingData | 调用者提供的资格数据 |
| TCMT_SIG_SCHEME+ | inScheme | 如果 signHandle 的签名策略为 TCM2_ALG_NULL,使用该值作为签名策略 |
| TCML_PCR_SELECTION | PCRselect | quote 操作使用的 PCR |

**表 70　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 应该是 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 |

### 6.11.4　TCM2_GetTime

返回当前的时间。

该命令接口输入和输出参数及说明见表 71 和表 72。

**表 71　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)<br>授权索引:1<br>授权角色:USER |

表 71　TCM2_GetTime()接口输入参数（续）

| 类型 | 名称 | 描述 |
| --- | --- | --- |
| TCMI_DH_OBJECT+ | @signHandle | 能执行签名操作的密钥句柄<br>授权索引:2<br>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 72　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 应是 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 |

## 6.12　临时 EC 密钥命令

### 6.12.1　TCM2_EC_Ephemeral

TCM2_EC_Ephemeral 临时 EC 密钥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参数及说明见表 73 和表 74。

表 73　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 | 用于计算临时密钥的曲线 |

表 74　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 应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |

表 74　TCM2_EC_Ephemeral()接口输出参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2_RC | responseCode | 返回码 |
| TCM2B_ECC_POINT | Q | 临时公钥 $Q:=[r]G$ |
| UINT16 | counter | commitCount 的低 16 位 |

### 6.12.2　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 参数应是有签名属性(TCM2_RC_ATTRIBUTES)的 ECC 密钥，签名方案必须是匿名的(TCM_RC_scheme)。

TCM_ALG_ECDAA 是唯一定义的匿名方案

注 1：此命令不能与 sign+decrypt 密钥一起使用，因为该类型的密钥必须具有 TCM_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，则 TCM 应返回 TCM_RC_SIZE。

该命令接口输入和输出参数及说明见表 75 和表 76。

表 75　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<br>授权索引：1<br>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 $x$ 坐标的八位数组 |
| TCM2B_ECC_PARAMETER | y2 | 与 s2 相关点的 $y$ 坐标 |

表 76　TCM2_Commit()接口输出参数

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM_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 |
| TCM_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 位 |

## 6.13 签名及签名验证命令

### 6.13.1 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。

验证签名遵循 GB/T 32918(所有部分)。

该命令接口输入和输出参数及说明见表 77 和表 78。

**表 77 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_SESSION，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<br>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 78 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，则输出标记应是 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 |

### 6.13.2 TCM2_Sign

该命令使 TCM 用非对称签名密钥对外部提供的杂凑值进行签名。

如果 keyHandle 引用受限签名密钥，则应提供验证，表明 TCM 执行了杂凑计算，验证应表明杂凑数据不是以 TCM2_GENERATED_VALUE 开始的。

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。

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。

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。

签名遵循 GB/T 32918(所有部分)。

该命令接口输入和输出参数及说明见表 79 和表 80。

**表 79　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<br>授权索引:1<br>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 | 证明摘要值是由 TCM 产生的凭证信息 |

**表 80　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,则输出标记应是 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 |

## 6.14　度量命令

### 6.14.1　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 绑定的摘要值。对每个摘要值,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]:=Halg(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参数及说明见表 81 和表 82。

表 81　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} |
| TCMI_DH_PCR+ | @pcrHandle | PCR 句柄<br>授权句柄:1<br>授权角色:USER |
| TCML_DIGEST_VALUES | digests | 待扩展的摘要值列表 |

表 82　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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.14.2　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参数及说明见表 83 和表 84。

表 83　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 84　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 |

### 6.14.3　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，那么该命令可以用于设置 PCR 的值为零。PCR 的属性会限制执行重置操作的位置(locality)。

如果 pcrHandle 参数指向的 PCR 不能执行重置，那么 TCM 将会返回 TCM2_RC_LOCALITY。

该命令接口输入和输出参数及说明见表 85 和表 86。

表 85　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<br>授权索引:1<br>授权角色:USER |

表 86　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

## 6.15　增强授权命令

### 6.15.1　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 是一个试验会话(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 不是试验会话,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:=HauthAlg(nonceTCM||expiration||cpHashA||policyRef)

该命令接口输入和输出参数及说明见表 87 和表 88。

**表 87 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<br>授权索引:None |
| TCMI_SH_POLICY | policySession | 扩展的策略会话句柄<br>授权索引:None |
| TCM2B_NONCE | nonceTCM | 会话的策略 nonce 值<br>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<br>该值不是命令需要的 cpHash,而是策略会话被应用的命令的 cpHash 值,如果没有限制,该参数可以为空 |
| TCM2B_NONCE | policyRef | 指向于授权相关的策略(可以为空)<br>大小不能大于 TCM 支持的 nonce 的尺寸大小 |
| UINT32 | expiration | 授权将过期的时间,从 nonceTCM 创建开始以秒为单位进行计时<br>如果 expiration 是零,没有票据返回 |
| TCMT_SIGNATURE | auth | 签名的授权(非可选) |

**表 88 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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

表 88　TCM2_PolicySigned()接口输出参数（续）

| 类型 | 名称 | 描述 |
| --- | --- | --- |
| TCM2B_TIMEOUT | timeout | 执行特定时间值，当票据过期时用于指向 TCM<br>注：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 |

### 6.15.2　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需要满足该需求。

如果使用了策略会话，但是 authHandle 的 authValue 不是需要的，TCM 将会返回 TCM2_RC_MODE。

该命令接口输入和输出参数及说明见表 89 和表 90。

表 89　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<br>授权句柄：1<br>授权角色：USER |
| TCMI_SH_POLICY | policySession | 被扩展策略会话的句柄<br>授权索引：None |
| TCM2B_NONCE | nonceTCM | 会话的策略 nonce<br>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<br>该值不是命令需要的 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 | 指向与授权相关的策略（可为空） |
| UINT32 | expiration | 授权将过期的时间，从 nonceTCM 创建开始以秒为单位进行计时<br>如果 expiration 是零，没有票据返回 |

表 90　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，则输出标记应是 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<br>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 |

### 6.15.3　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参数及说明见表 91 和表 92。

表 91　TCM2_PolicyTicket()接口输入参数

| 类型 | 名称 | 描述 |
|---|---|---|
| TCMI_ST_COMMAND_TAG | tag | 如果是解密会话，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<br>授权索引：None |
| TCM2B_TIMEOUT | timeout | 授权将要过期的时间<br>内容由 TCM 指定，可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<br>该值不是命令需要的 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 |

表 91　TCM2_PolicyTicket()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2B_NONCE | policyRef | 指向于授权相关的策略(可为空)<br>大小不能大于 TCM 支持的 nonce 的尺寸大小 |
| TCM2B_NAME | authName | 提供授权对象的名称 |
| TCMT_TK_AUTH | ticket | TCM2_PolicySigned()或 TCM2_PolicySecret()返回的 TCM 授权票据 |

表 92　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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.15.4　TCM2_PolicyOR

该命令是 TCM 不需要对所有的选项进行评估,就选择授权选项。如果一个策略可能通过不同条件集满足,TCM 只需要评估一个集合是满足该策略的即可。该命令将会显示出其中一个被需要的条件集合已经满足。

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 是一个试验会话,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参数及说明见表 93 和表 94。

表 93　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<br>授权索引:None |
| TCML_DIGEST | pHashList | 匹配检测的杂凑列表 |

表 94　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 |

### 6.15.5　TCM2_PolicyPCR

该命令用于触发基于 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。如果命令用于试验策略会话，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，以便清除与未实现的 PCR 相对应的位。如果 policySession 不是试验策略会话，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，需要比较 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参数及说明见表 95 和表 96。

表 95　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<br>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摘要值<br>可能为零长度 |
| TCML_PCR_SELECTION | pcrs | 包含在检测摘要中的 PCR |

表 96　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.15.6 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命令码的值，当 policySession->commandCode 没有默认值，如果两值不同时，TCM 将会返回 TCM2_RC_VALUE。

如果命令码没有实现，TCM 将会返回 TCM2_RC_POLICY_CC。如果 TCM 没有返回错误，policySession->policyDigest 将会采用如下值被更新：

policyDigestnew：=HpolicyAlg(policyDigestold||TCM2_CC_PolicyCommandCode||code)

当策略会话用于授权命令，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参数及说明见表 97 和表 98。

**表 97 TCM2_PolicyLCommandCode()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<br>授权索引：None |
| TCM2_CC | code | 允许的命令代码 |

**表 98 TCM2_PolicyLCommandCode()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 |

### 6.15.7 TCM2_PolicyPhysicalPresence

该命令表明在执行授权时需要声明物理存在。

如果该命令成功，policySession->isPPRequired 将会被设置，这表明该策略用于授权时必须被检测。另外，policySession->policyDigest 使用如下值进行扩展：

policyDigestnew：=HpolicyAlg(policyDigestold||TCM2_CC_PolicyPhysicalPresence)

该命令接口输入和输出参数及说明见表 99 和表 100。

**表 99 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 |

表 99　TCM2_PolicyPhysicalPresence()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2_CC | commandCode | TCM2_CC_PolicyPhysicalPresence |
| TCMI_SH_POLICY | policySession | 被扩展策略会话的句柄<br>授权索引：None |

表 100　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 |

### 6.15.8　TCM2_PolicyCpHash

该命令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 被设计于可以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以授权特定命令代码。

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：

policyDigestnew：=HpolicyAlg(policyDigestold||TCM2_CC_PolicyCpHash||cpHashA)

该命令接口输入和输出参数及说明见表 101 和表 102。

表 101　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<br>授权索引：None |
| TCM2B_DIGEST | cpHashA | 加到策略的 cpHash |

表 102　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 应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.15.9　TCM2_PolicyAuthValue

该命令允许将策略绑定到授权对象的授权值。

当该命令成功完成，policySessin-〉isAuthValueNeeded 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。

操作如果成功，policySession-〉policyDigest 将会使用如下值更新：

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

该命令接口输入和输出参数及说明见表 103 和表 104。

表 103　TCM2_PloicyAuthValue()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<br>授权索引：None |

表 104　TCM2_PloicyAuthValue()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 |

### 6.15.10　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参数及说明见表 105 和表 106。

表 105　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<br>授权索引:None |

表 106　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 |

### 6.15.11　TCM2_PolicyGetDigest

该命令返回当前会话的 policyDigest 的值。

该命令允许 TCM 用于提前为对象计算 authPolicy 值。

该命令接口输入和输出参数及说明见表 107 和表 108。

表 107　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 |
| TCM2_CC | commandCode | TCM2_CC_PolicyGetDigest |
| TCMI_SH_POLICY | policySession | 被扩展策略会话的句柄<br>授权索引:None |

表 108　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,则输出标记应是 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 |

## 6.16 分层命令

### 6.16.1 TCM2_CreatePrimary

该命令用于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。

该命令可以创建 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 相关联的主对象要求提供 platformAuth 或 platformPolicy 授权信息，与 SPS 相关联的主要对象要求调用者提供 ownerAuth 或 ownerPolicy 授权信息，与 EPS 相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参数及说明见表 109 和表 110。

**表 109 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，<br>TCM2_RH_OWNER，<br>TCM2_RH_PLATFORM+{PP}或者 TCM2_RH_NULL<br>授权索引：1<br>授权角色：USER |
| TCM2B_SENSITIVE_CREATE | inSensitive | 隐私数据部分 |
| 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 110 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，则输出标记应是 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 |

表 110　TCM2_CreatePrimary()接口输出参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| 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 |

### 6.16.2　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相关联的永久实体，且将清除所有与该等级相关联的临时对象。

该命令接口输入和输出参数及说明见表 111 和表 112。

表 111　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，<br>TCM2_RH_OWNER 或者 TCM2_RH_PLATFORM＋{PP}<br>授权索引：1<br>授权身份：USER |
| TCMI_RH_ENABLES | enable | 将被改变的 hierarchy 等级<br>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 112　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 应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.16.3 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，该会话应该使用当前的授权值，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，那么相关联的授权信息将不被使用。

该命令接口输入和输出参数及说明见表 113 和表 114。

**表 113 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 | @authHandle | TCM2_RH_ENDORSEMENT，TCM2_RH_OWNER 或者 TCM2_RH_PLATFORM+{PP}<br>授权索引：1<br>授权身份：USER |
| TCM2B_DIGEST | authPolicy | 授权策略的摘要值，如果 hashAlg 参数为空，则此参数应该为 Empty Buffer。 |
| TCMI_ALG_HASH+ | hashAlg | 计算摘要使用的 hash 算法，如果参数 authPolicy 为空，则此参数应该为 TCM2_ALG_NULL。 |

**表 114 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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据长度 |
| TCM2_RC | responseCode | 返回码 |

### 6.16.4 TCM2_Clear

此命令移除与指定所有者相关联的所有 TCM 上下文。

此命令要求 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参数及说明见表 115 和表 116。

表 115　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}<br>授权索引:1<br>授权身份:USER |

表 116　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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.16.5　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 授权信息可以禁用 TCM2_Clear 命令但是不能启用。使用 PlatformAuth 授权信息可以禁用或启用 TCM2_Clear 命令。使用 LockoutAuth 授权用于设置 disableClear,但不能清除。使 LatformAuth 授权用于设置或禁用清除。

该命令接口输入和输出参数及说明见表 117 和表 118。

表 117　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{NV} |
| TCMI_RH_CLEAR | @auth | TCM2_RH_LOCKOUT or TCM2_RH_PLATFORM+{PP}<br>授权索引:1<br>授权身份:USER |
| TCMI_YES_NO | disable | YES:要设置 disableOwherClear 标志<br>NO:清除 disableOwnerClear 标志 |

表 118　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据长度 |
| TCM2_RC | responseCode | 返回码 |

### 6.16.6　TCM2_HierarchyChangeAuth

该命令将改变一个 hierarchy 等级或 lockout 的授权秘密值，需要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杂凑算法产生的摘要值长度。

该命令接口输入和输出参数及说明见表 119 和表 120。

表 119　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，<br>TCM2_RH_ENDORSEMENT，<br>TCM2_RH_OWNER 或<br>TCM2_RH_PLATFORM+{PP}<br>授权索引：1<br>授权身份：USER |
| TCM2B_AUTH | newAuth | 新的授权值 |

表 120　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 |

## 6.17　字典攻击命令

### 6.17.1　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参数及说明见表 121 和表 122。

**表 121　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<br>授权索引：1<br>授权角色：USER |

**表 122　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.17.2　TCM2_DictionaryAttackParameters

该命令用于改变锁定参数。

该命令需要 lockoutAuth。

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。

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参数及说明见表 123 和表 124。

**表 123　表 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<br>授权索引：1<br>授权角色：USER |
| UINT32 | newMaxTries | 被锁定之前支持的最大失败次数 |

表 123　表 TCM2_DictionaryAttackParameters()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| 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<br>零值表示需要重启 |

表 124　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

## 6.18　管理功能命令

TCM2_PP_Commands 命令用于定义作为对 platformAuth/platformPolicy 的补充，哪些命令需要声明物理存在(PP)。

该命令需要 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当与句柄关联的授权是 TCM2_RH_PLATFORM 时需要声明物理存在的命令列表中。clearList 列表中的命令不再需要声明物理存在。

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，它将不再需要声明物理存在。

TCM2_PP_Command()命令总是需要声明物理存在。

该命令接口输入和输出参数及说明见表 125 和表 126。

表 125　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} |
| TCMI_RH_PLATFORM | @auth | TCM2_RH_PLATFORM+PP<br>授权索引：1<br>授权角色：USER+Physical Presence |
| TCML_CC | setList | 增加需要物理存在声明的命令列表 |
| TCML_CC | clearList | 不再需要物理存在声明的命令列表 |

表 126　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

## 6.19　上下文管理命令

### 6.19.1　TCM2_ContextSave

该命令用于保存会话上下文或序列对象上下文。

该命令不需要任何需要授权的会话，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参数及说明见表 127 和表 128。

表 127　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 | 被保存的资源<br>授权索引：None |

表 128　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 |

### 6.19.2　TCM2_ContextLoad

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

该命令不需要任何需要授权的会话，tag 应是 TCM2_ST_NO_SESSIONS。

当与上下文关联的分层管理是禁止的，TCM 将会返回 TCM2_RC_HIERARCHY。

该命令接口输入和输出参数及说明见表 129 和表 130。

表 129　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 130　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 |

### 6.19.3　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。

该命令不需要任何需要授权的会话，tag 必须是 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参数及说明见表 131 和表 132。

表 131　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 132　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 |

### 6.19.4 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参数及说明见表 133 和表 134。

**表 133 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}<br>授权句柄：1<br>授权角色：USER |
| TCMI_DH_OBJECT | objectHandle | 被加载对象句柄<br>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<br>如果 objectHandle 是持久对象句柄，那么该值与 objectHandle 值一致 |

**表 134 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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

## 6.20 性能命令

### 6.20.1 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 值相关的特性值，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需要的功能数量。TCM 将会返回需要数量的被请求的值(propertyCount)或者返回被需要类型的所有的特性。

当被需要所有特性已被返回，moreData 参数设置为 NO，否则设置为 YES。

该命令接口输入和输出参数及说明见表 135 和表 136。

表 135　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,确定了返回值得格式 |
| UINT32 | property | 属性 |
| UINT32 | propertyCount | 指明类型返回的属性数量 |

表 136　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 | capability 数据 |

### 6.20.2　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参数及说明见表 137 和表 138。

表 137　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 138　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 |

## 6.21 NV 操作命令

### 6.21.1 TCM2_NV_DefineSpace

该命令为 NV 索引定义属性，TCM 预留用于保留与 NV 索引相关数据的空间。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。

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。

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。

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。

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参数及说明见表 139 和表 140。

**表 139 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}<br>授权索引：1<br>授权角色：USER |
| TCM2B_AUTH | auth | 授权值 |
| TCM2B_NV_PUBLIC | publicInfo | NV 区的公共参数 |

**表 140 TCM2_NV_DefineSpac()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 |

### 6.21.2 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参数及说明见表 141 和表 142。

**表 141　TCM2_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}<br>授权索引：1<br>授权角色：USER |
| TCMI_RH_NV_INDEX | nvIndex | 从 NV 空间移除的 NV 索引<br>授权索引：None |

**表 142　TCM2_UndefieSpace()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 应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.3　TCM2_NV_ReadPublic

该命令用于读取给定 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，所以读取这些数据时不需要授权。

该命令接口输入和输出参数及说明见表 143 和表 144。

**表 143　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<br>授权索引：None |

表 144　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 |

### 6.21.4　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。

该命令的使用需要使用正确的授权，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参数及说明见表 145 和表 146。

表 145　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 {NV} |
| TCMI_RH_NV_AUTH | @authHandle | 指向授权值源的句柄<br>授权索引：1<br>授权角色：USER |
| TCMI_RH_NV_INDEX | nvIndex | 待写 NV 区的索引<br>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 146　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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.5 TCM2_NV_Increment

该命令用于设置了 TCMA_NV_COUNTER 属性的 NV 索引中的值。NV 索引的数据值递增加 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参数及说明见表 147 和表 148。

**表 147 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 {NV} |
| TCMI_RH_NV_AUTH | @authHandle | 指向授权值源的句柄<br>授权索引：1<br>授权角色：USER |
| TCMI_RH_NV_INDEX | nvIndex | 待增加的 NV 索引<br>授权索引：None |

**表 148 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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.6 TCM2_NV_Extend

该命令对使用 TCM2_NV_DefineSpace 命令定义的 NV 区域进行扩展值。

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 的 TCMA_NV_WRITELOCKED 属性被设置，TCM 将会返回 TCM2_RC_NV_LOCKED。

data.buffer 参数可以大于 NV 索引定义的 NV 空间大小。

该命令接口输入和输出参数及说明见表 149 和表 150。

**表 149 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 |

表 149　TCM2_NV_Extend()接口输入参数（续）

| 类型 | 名称 | 描述 |
|---|---|---|
| TCM2_CC | commandCode | TCM2_CC_NV_Extend {NV} |
| TCMI_RH_NV_AUTH | @authHandle | 指向授权值源的句柄<br>授权索引:1<br>授权角色:USER |
| TCMI_RH_NV_INDEX | nvIndex | 待扩展的 NV 索引<br>授权索引:None |
| TCM2B_MAX_NV_BUFFER | data | 待扩展到 NV 区的数据 |

表 150　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 应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.7　TCM2_NV_SetBits

该命令用于设置 NV 索引中的比特位。任意数量从 0 到 64 的比特位可以被设置。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位,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参数及说明见表 151 和表 152。

表 151　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} |
| TCMI_RH_NV_AUTH | @authHandle | 指向授权值源的句柄<br>授权索引:1<br>授权角色:USER |
| TCMI_RH_NV_INDEX | nvIndex | 需设置比特位 NV 区的 NV 索引<br>授权索引:None |
| UINT64 | bits | 与当前内容进行或操作的数据 |

表 152　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 应该是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.8　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。

该命令需要使用正确的写授权。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参数及说明见表 153 和表 154。

表 153　TCM2_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<br>授权索引：1<br>授权角色：USER |
| TCMI_RH_NV_INDEX | nvIndex | 待锁定 NV 区的 NV 索引<br>授权索引：None |

表 154　TCM2_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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.9 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，该命令将永久锁定该 NV 索引的写操作。

该命令需要 platformAuth/platformPolicy 或者 ownerAuth/ownerPolicy 授权。

该命令接口输入和输出参数及说明见表 155 和表 156。

**表 155 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}<br>授权索引：1<br>授权角色：USER |

**表 156 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，则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.10 TCM2_NV_Read

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

该命令的使用需要使用正确的授权，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参数及说明见表 157 和表 158。

**表 157 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 |

表 157　TCM2_NV_Read()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<br>授权索引:1<br>授权角色:USER |
| TCMI_RH_NV_INDEX | nvIndex | 执行读操作的 NV 索引<br>授权索引:None |
| UINT16 | size | 准备读的字节数 |
| UINT16 | offset | NV 区的偏移量<br>该值可以小于或者等于 nvIndex 的数据大小 |

表 158　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,则输出标记应是 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 |

### 6.21.11　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)。

该命令需要正确的授权。如定义的 TCMA_NV_PPREAD,TCMA_NV_OWNERREAD,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参数及说明见表 159 和表 160。

表 159　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 |

表 159　TCM2_NV_ReadLock()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<br>授权索引:1<br>授权角色:USER |
| TCMI_RH_NV_INDEX | nvIndex | 待锁定的 NV 区的 NV 索引<br>授权索引:None |

表 160　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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

### 6.21.12　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参数及说明见表 161 和表 162。

表 161　TCM2_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{NV} |
| TCMI_RH_NV_INDEX | @nvIndex | 对象句柄<br>授权索引:1<br>授权角色:ADMIN |
| TCM2B_AUTH | newAuth | 新的授权值 |

表 162　TCM2_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,则输出标记应是 TCM2_ST_NO_SESSIONS |
| UINT32 | responseSize | 返回数据大小 |
| TCM2_RC | responseCode | 返回码 |

# 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 |

表 A.1　命令码（续）

| 名称 | 命令码 |
| --- | --- |
| TCM2_CC_DictionaryAttackLockReset | 0x00000139 |
| TCM2_CC_DictionaryAttackParameters | 0x0000013A |
| 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 |
| TCM2_CC_StirRandom | 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 |

表 A.1 命令码（续）

| 名称 | 命令码 |
| --- | --- |
| TCM2_CC_Quote | 0x00000158 |
| TCM2_CC_ECC_Decrypt | 0x00000159 |
| TCM2_CC_HMAC_Start | 0x0000015B |
| TCM2_CC_SequenceUpdate | 0x0000015C |
| TCM2_CC_Sign | 0x0000015D |
| 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 |
| 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 |

表 A.1　命令码（续）

| 名称 | 命令码 |
| --- | --- |
| 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 |
| 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_EncryptDecrypt2 | 0x00000193 |
| Reserved | 0x00000194 |
| Reserved | 0x00000195 |
| Reserved | 0x00000196 |
| Reserved | 0x00000197 |
| Reserved | 0x00000198 |
| TCM2_CC_ECC_Encrypt | 0x00000199 |
| TCM2_CC_Decrypt | 0x0000019A |
| TCM2_CC_LAST | 0x0000019A |
| CC_VEND | 0x20000000 |
| 注：文中“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

返回码见表 A.2。

**表 A.2 返回码**

| 名称 | 值 |
|---|---|
| TCM2_RC_SUCCESS | 0x000 |
| TCM2_RC_BAD_TAG | 0x01E |
| 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 |

表 A.2　返回码（续）

| 名称 | 值 |
| --- | --- |
| 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 |
| 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 |

表 A.2　返回码（续）

| 名称 | 值 |
| --- | --- |
| 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 |
| 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 |

表 A.2　返回码（续）

| 名称 | 值 |
| --- | --- |
| 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 |
| 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

表 A.3 定义了基本类型。

**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

表 A.4 定义了逻辑数值常量。

**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

表 A.5 对其他类型进行了定义。

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)

表 A.6 为 TCM2_SPEC 常量定义,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 | 2020 | 年份版本 |
| Reserved | | |

#### A.3.4.2 TCM2_ALG_ID

从表 A.7～表 A.10 对 TCM2_ALG_ID 图例及 ID 进行了定义。

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:<br>含有公钥和私钥的非对称算法<br>S——仅有一个私钥的对称算法<br>H——将输入数据压缩成摘要值的杂凑算法,也可声明为使用了杂凑算法<br>X——签名算法<br>N——一个匿名签名算法<br>E——一个加密算法<br>M——一个如 Mask 产生功能的方法<br>O——一个对象类型 |

表 A.7　TCM2_ALG_ID 表图例（续）

| 列名称 | 描述 |
| --- | --- |
| 依赖 | 如果这个算法被声明，表明还需要其他算法需要被声明 |
| 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 | | | |
| TCM2_ALG_ECDAA | 0x001A | A,X,N | ECC | |

表 A.9　TCM2_ALG_ID(UINT16)定义

| 算法名称 | 值 | 类型 | 依赖 | 描述 |
| --- | --- | --- | --- | --- |
| TCM2_ALG_SM2 | 0x001B | A,X | ECC | 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 | 0X0024 | | | |

表 A.9　TCM2_ALG_ID(UINT16)定义（续）

| 算法名称 | 值 | 类型 | 依赖 | 描述 |
|---|---|---|---|---|
| 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 |

表 A.10　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 | | |

### A.3.4.3　TCM2_CC(命令码)

一个命令是一个 32 位结构,其字段被分配,如表 A.11 和表 A.12 所示。

表 A.11　TCM 命令 32 位结构

| 31 | 30 | 29 | 28 | | | | | | | | | | | | 16 | 15 | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res | | V | Reserved | | | | | | | | | | | | | 命令索引 | | | | | | | | | | | | | | |

表 A.12　TCM 命令格式字段描述

| 比特位 | 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<br>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)

表 A.13 对 TCM2_ST(UINIT16)进行了定义。

表 A.13　TCM2_ST(UINT16)定义

| 名称 | 值 | 描述 |
|---|---|---|
| TCM2_ST_RSP_COMMAND | 0x00C4 | 返回值的标识<br>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<br>应用的命令/返回值不涉及会话,无授权大小/参数<br>如果 TCM 的返回码(responseCode)不是 |
| TCM2_ST_SESSIONS | 0x8002 | 命令/返回的标识的值<br>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 |
| TCM2_ST_ATTEST_COMMAND_AUDIT | 0x8015 | 标识用于认证的结构体 |
| TCM2_ST_ATTEST_SESSION_AUDIT | 0x8016 | 标识用于认证的结构体 |
| 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 |

### A.3.4.5　TCM2_CAP(Capilities)

表 A.14 对 TCM2_CAP 的值进行了定义。

表 A.14　TCM2_CAP 值

| 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 |

表 A.14 TCM2_CAP 值（续）

| 名称 | 值 | 属性 | 返回类型 |
|---|---|---|---|
| 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）

表 A.15 对 TCM2_PT 值进行了定义。

表 A.15 TCM2_PT 值

| 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 | 组中固定属性返回 TCMS_TAGGED_PROPERTY<br>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<br>例如，修订版本号位 01.01，结果会是 101 |
| TCM2_PT_DAY_OF_YEAR | PT_FIXED+3 | 规范的日期<br>例如 11/15/2010 为一年中的第 319 天，值为 319(00 00 01 3F) |
| TCM2_PT_YEAR | PT_FIXED+4 | 规范使用的年份<br>例：2010 年为 00 00 07 DA |
| TCM2_PT_MANUFACTURER | PT_FIXED+5 | 每个厂商的统一 ID |

表 A.15　TCM2_PT 值（续）

| 名称 | 值 | 描述 |
|---|---|---|
| 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=1 024） |
| 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<br>至少为 $2^{16}-1$(65 535). |
| | PT_FIXED+21 | 放弃 |
| 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 |
| TCM2_PT_CONTEXT_HASH | PT_FIXED+26 | 该算法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.15 TCM2_PT 值（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 |

### A.3.4.7 TCM2_PT_PCR（PCR 属性标识）

表 A.16 对 TCM2_PT_PCR(UINT32)进行了定义。

表 A.16 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 一个比特位被设置，用于表明 PCR 被 TCM2_SU_STATE 保存和恢复 |
| TCM2_PT_PCR_EXTEND_L0 | 0x00000001 | TCMS_PCR_SELECT 一个比特位被设置，表明 PCR 从 locality 0 扩展 |
| TCM2_PT_PCR_RESET_L0 | 0x00000002 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 TCM2_PCR_Reset()从 locality 0 被重置 |
| TCM2_PT_PCR_EXTEND_L1 | 0x00000003 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是从 locality 1 被扩展<br>仅当 locality 1 被声明时这个属性才出现 |
| TCM2_PT_PCR_RESET_L1 | 0x00000004 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 TCM2_PCR_Reset()从 locality 1 被重置<br>仅当 locality 1 被声明时这个属性才出现 |
| TCM2_PT_PCR_EXTEND_L2 | 0x00000005 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是从 locality 2 被扩展<br>仅当 locality 1 和 locality 2 被声明时这个属性才出现 |

表 A.16　TCM2_PT_PCR(UINT32)定义（续）

| 名称 | 值 | 描述 |
| --- | --- | --- |
| TCM2_PT_PCR_RESET_L2 | 0x00000006 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 TCM2_PCR_Reset()从 locality 2 被重置<br>仅当 locality 1 和 locality 2 被声明时这个属性才出现 |
| TCM2_PT_PCR_EXTEND_L3 | 0x00000007 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是从 locality 3 被扩展<br>仅当 locality 1、locality 2 和 locality 3 被声明时这个属性才出现 |
| TCM2_PT_PCR_RESET_L3 | 0x00000008 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 TCM2_PCR_Reset()从 locality 3 被重置<br>仅当 locality 1、locality 2 和 locality 3 被声明时这个属性才出现 |
| TCM2_PT_PCR_EXTEND_L4 | 0x00000009 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是从 locality 4 被扩展<br>仅当 locality 1、locality 2、locality 3 和 locality 4 被声明时这个属性才出现 |
| TCM2_PT_PCR_RESET_L4 | 0x0000000A | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 TCM2_PCR_Reset()从 locality 4 被重置<br>仅当 locality 1、locality 2、locality 3 和 locality 4 被声明时这个属性才出现 |
| reserved | 0x0000000B～<br>0x00000010 | 保留 |
| TCM2_PT_PCR_NO_INCREMENT | 0x00000011 | TCMS_PCR_SELECT 一个比特位设置，表明修改 PCR(重置或者扩展)将不会使 pcrUpdateCounter 增加 |
| TCM2_PT_PCR_DRTM_RESET | 0x00000012 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被 D-RTM 事件重置的 |
| TCM2_PT_PCR_POLICY | 0x00000013 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被策略控制的 |
| TCM2_PT_PCR_AUTH | 0x00000014 | TCMS_PCR_SELECT 一个比特位设置，表明 PCR 是被授权值控制的 |
| reserved | 0x00000015 | 保留 |
| reserved | 0x00000016 | 保留 |
| reserved | 0x00000017～<br>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)

表 A.17 对句柄进行了定义,表 A.18 对常量进行了定义。

**表 A.17 句柄定义**

| 类型 | 名称 | 描述 |
|---|---|---|
| UINT32 | TCM2_HANDLE | |

**表 A.18 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 PCR 寄存器<br>一个平台需要设置一个最小的 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<br>涉及加载 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<br>这个涉及为 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)

表 A.19 对 TCM2_RH(TCM2_HANDLE)进行了定义。

**表 A.19 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 |

表 A.19　TCM2_RH(TCM2_HANDLE)定义（续）

| 名称 | 值 | 类型 | 描述 |
|---|---|---|---|
| TCM2_RH_OPERATOR | 0x40000004 | R | 保留 |
| TCM2_RH_ADMIN | 0x40000005 | R | 保留 |
| TCM2_RH_EK | 0x40000006 | R | 保留 |
| TCM2_RH_NULL | 0x40000007 | K,A,P | 用于非空 hierarchy,an 空授权的 authValue,and 空策略的 authPolicy 的句柄 |
| TCM2_RH_UNASSIGNED | 0x40000008 | R | 保留 |
| TCM2_RS_PW | 0x40000009 | S | 用于声明密码授权绘画的授权值 |
| 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),怕平台授权(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：<br>R ——保留值<br>K ——原始种子<br>A ——授权值<br>P ——策略值<br>S ——会话 handle<br>C ——控制 | | | |

### A.3.5.3　TCM2_HC(Handle 值常量)

表 A.20 对 TCM2_HC(TCM2_HANDLE)进行了定义。

表 A.20　TCM2_HC(TCM2_HANDLE)定义

| 名称 | 值 | 说明 |
|---|---|---|
| HR_HANDLE_MASK | 0x00FFFFFF | |
| HR_RANGE_MASK | 0xFF000000 | |
| HR_SHIFT | 24 | |

表 A.20　TCM2_HC(TCM2_HANDLE)定义（续）

| 名称 | 值 | 说明 |
| --- | --- | --- |
| 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) | |
| 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 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_FIRST + MAX_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

表 A.21 为 TCMI_YES_NO(BYTE)的定义。

**表 A.21 TCMI_YES_NO(BYTE)定义**

| 类型 | 名称 |
|---|---|
| NO | 0 |
| YES | 1 |
| ＃TCM2_RC_VALUE | |

#### A.3.6.2 TCMI_DH_OBJECT

表 A.22 为 TCMI_DH_OBJECT 的定义。

TCMI_DH_OBJECT 对象接口类型是引用加载对象的句柄。此集中的句柄用于引用临时对象或持久对象。

**表 A.22 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

表 A.23 为 TCMI_DH_PARENT 的定义。

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。

**表 A.23 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

表 A.24 为 TCMI_DH_PERSISTENT 的定义。

**表 A.24 TCMI_DH_PERSISTENT 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 TCMI_DH_ENTITY

表 A.25 为 TCMI_DH_ENTITY 的定义。

**表 A.25 TCMI_DH_ENTITY 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

表 A.26 为 TCMI_DH_PCR 的定义。

**表 A.26 TCMI_DH_PCR 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 TCMI_SH_AUTH_SESSION

表 A.27 为 TCMI_SH_AUTH_SESSION 的定义，见表 A.26。

**表 A.27 TCMI_SH_AUTH_SESSION 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 TCMI_SH_HMAC

表 A.28 为 TCMI_SH_HMAC 的定义。

**表 A.28 TCMI_SH_HMAC 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 TCMI_SH_POLICY

表 A.29 为 TCMI_SH_POLICY 的定义。

**表 A.29 TCMI_SH_POLICY 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

表 A.30 为 TCMI_DH_CONTEXT 的定义。

**表 A.30 TCMI_DH_CONTEXT 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

表 A.31 为 TCMI_RH_HIERARCHY 的定义。

**表 A.31 TCMI_RH_HIERARCH 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

表 A.32 为 TCMI_SH_ENABLES 的定义。

**表 A.32 TCMI_SH_ENABLES 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

表 A.33 为 TCMI_HIERARCHY_AUTH 的定义。

**表 A.33 TCMI_HIERARCHY_AUTH 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

表 A.34 为 TCMI_RH_PLATFORM 的定义。

**表 A.34 TCMI_RH_PLATFORM 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 TCMI_RH_OWNER

表 A.35 为 TCMI_RH_OWNER 的定义。

**表 A.35 TCMI_RH_OWNER 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

表 A.36 为 TCMI_RH_ENDORSEMENT(TCM2_HANDLE)的定义。

**表 A.36 TCMI_RH_ENDORSEMENT(TCM2_HANDLE)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

表 A.37 为 TCM1_RH_PROVISION(TCM2_HANDLE)的定义。

**表 A.37 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

表 A.38 为 TCM1_RH_CLEAR(TCM2_HANDLE)的定义。

**表 A.38 TCMI_RH_CLEAR(TCM2_HANDLE)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

表 A.39 为 TCM1_RH_NV_AUTH(TCM2_HANDLE)的定义。

**表 A.39 TCMI_RH_NV_AUTH(TCM2_HANDLE)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

表 A.40 为 TCM1_RH_LOCKOUT(TCM2_HANDLE)的定义。

**表 A.40 TCMI_RH_LOCKOUT(TCM2_HANDLE)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

表 A.41 为 TCM1_RH_NV_INDEX(TCM2_HANDLE)的定义。

**表 A.41 TCMI_RH_NV_INDEX(TCM2_HANDLE)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

表 A.42 为 TCM1_ALG_HASH(TCM2_ALG_ID)的定义。

**表 A.42 TCMI_ALG_HASH(TCM2_ALG_ID)定义**

| 值 | 描述 |
|---|---|
| TCM2_ALG_! ALG.H | 国家密码管理局定义 hash 算法 |
| +TCM2_ALG_NULL | |
| #TCM2_RC_HASH | |

### A.3.6.23 TCMI_ALG_ASYM(非对称算法)

表 A.43 为 TCM1_ALG_ASYM(TCM2_ALG_ID)的定义。

**表 A.43 TCMI_ALG_ASYM(TCM2_ALG_ID)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)

表 A.44 为 TCM1_ALG_SYM(TCM2_ALG_ID)的定义。

**表 A.44 TCMI_ALG_SYM(TCM2_ALG_ID)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

表 A.45 为 TCM1_ALG_SYM_OBJECT(TCM2_ALG_ID)的定义。

**表 A.45 TCMI_ALG_SYM_OBJECT(TCM2_ALG_ID)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

表 A.46 为 TCM1_ALG_SYM_MODE(TCM2_ALG_ID)的定义。

**表 A.46 TCMI_ALG_SYM_MODE(TCM2_ALG_ID)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_SIG_SCHEME

表 A.47 为 TCM1_ALG_SIG_SCHEME(TCM2_ALG_ID)的定义。

**表 A.47 TCMI_ALG_SIG_SCHEME(TCM2_ALG_ID)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.3.6.28 TCMI_ECC_KEY_EXCHANGE

表 A.48 为 TCMI_ECC_KEY_EXCHANGE(TCM2_ALG_ID)的定义。

**表 A.48 TCMI_ECC_KEY_EXCHANGE(TCM2_ALG_ID)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.3.6.29 TCMI_ST_COMMAND_TAG

表 A.49 为 TCM1_ST_COMMAND_TAG(TCM2_ST)的定义。

表 A.49 TCMI_ST_COMMAND_TAG(TCM2_ST)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 | |

## 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

表 A.50 为 TCMS_ALGORITHM_DESCRIPTION 结构定义，用于 TCM2_GetCapability()返回预装的算法。

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)

表 A.51 为 TCMU_HA 结构体定义，TCMU_HA 是 TCM 内部计算摘要的杂凑算法联合体。

表 A.51 TCMU_HA 结构体定义

| 参数 | 类型 | 选择 | 描述 |
|---|---|---|---|
| ！ALG.H［！ALG.H_DIGEST_SIZE］ | BYTE | TCM_ALG_！ALG.H | all hashes |
| null | | TCM2_ALG_NULL | |

#### A.4.3.2 TCMT_HA

表 A.52 为 TCMT_HA 结构体定义，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。

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

表 A.53 为 TCM2B_DIGEST 结构体定义，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。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

表 A.54 为 TCM2B_DATA 结构体定义，此结构用于对象名称大小的数据缓冲区。

**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

表 A.55 为 TCM2B_NONCE 结构体定义。

**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

表 A.56 为 TCM2B_AUTH 结构体定义，此结构用于授权值，并将 authValue 设置为 TCM 生成的最大摘要。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

表 A.57 为 TCM2B_OPERAND 结构体定义，此结构保存操作数以便与 NV 索引位置进行比较。

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

表 A.58 为 TCM2B_EVENT 结构体定义，此结构是事件数据的大小缓冲区。

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

表 A.59 为 TCM2B_MAX_BUFFER 结构体定义，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。

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 | MAX_DIGEST_BUFFER 最小为 1 024 |

### A.4.4.8　TCM2B_MAX_NV_BUFFER

表 A.60 为 TCM2B_MAX_NV_BUFFER 结构体定义，用于 NV 的最大缓冲压。

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<br>MAX_NV_BUFFER_SIZE 依赖 TCM |

### A.4.4.9　TCM2B_TIMEOUT

表 A.61 为 TCM2B_TIMEOUT 结构体定义，此结构用于提供授权的超时值。

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

表 A.62 为 TCM2B_IV 结构体定义，此结构用于传递对称分组密码的 IV 初始值。

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

表 A.63 为 TCMU_NAME 结构体定义。

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

表 A.64 为 TCM2B_NAME 结构体定义，此缓冲区保存任何实体类型的名称。

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。如果大小为零，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

表 A.65 为 TCMS_PCR_SELECT 结构体定义。

该结构提供了一种标准方法来指定 PCR 列表。PCR 编号从零开始。

pcrSelect 是一个八位字节数组。通过将 PCR 数除以 8,可找到包含与特定 PCR 对应的位的八位字节。

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

表 A.66 为 TCMS_PCR_SELECTION 结构体定义。

**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

表 A.67 为 Tickets 结构体定义。

**表 A.67 Tickets 结构体定义**

| 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.4.7.1 TCMS_TK_CREATION

表 A.68 为 TCMS_TK_CREATION 结构体定义，此票证由 TCM2_Create()或 TCM2_CreatePrimary()生成。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 {TCM_ST_CREATION} | TCM2_ST | 票据标签 |
| # TCM_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.4.7.2 TCMS_TK_VERIFIED

表 A.69 为 TCMS_TK_VERIFIED 结构体定义，此票证由 TCM2 验证签名()生成。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 {TCM_ST_VERIFIED} | TCM2_ST | 票据标签 |
| # TCM_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.4.7.3 TCMS_TK_AUTH

表 A.70 为 TCMS_TK_AUTH 结构体定义，当授权有过期时间时，此票证由 TCM2_PolicySigned()和 TCM2_PolicySecret()生成。

**表 A.70 TCMS_TK_AUTH 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| tag {TCM_ST_AUTH_SIGNED，TCM_ST_AUTH_SECRET} | TCM2_ST | 票据标签 |
| # TCM_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.4.7.4 TCMS_TK_HASHCHECK

表 A.71 为 TCMS_TK_HASHCHECK 结构体定义。

表 A.71 TCMS_TK_HASHCHECK 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| tag {TCM_ST_HASHCHECK} | TCM2_ST | 票据标签 |
| # TCM_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

表 A.72 为 TCMS_ALG_PROPERTY 结构体定义。

此结构用于报告算法标识符的属性。

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

表 A.73 为 TCMS_TAGGED_PROPERTY 结构体定义。

此结构用于报告 UINT32 值的属性。它是响应 TCM2_GetCapability()返回的。

表 A.73 TCMS_TAGGED_PROPERTY 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| property | TCM_PT | 属性标识符 |
| value | UINT32 | 属性值 |

#### A.4.8.3 TCMS_TAGGED_PCR_SELECT

表 A.74 为 TCMS_TAGGED_PCR_SELECT 结构体定义。

此结构在 TCM2_GetCapability()中用于返回 PCR 的属性。

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

表 A.75 为 TCMS_TAGGED_POLICY 结构体定义。

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

**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 | 策略算法与散列 |

### A.4.9 列表

#### A.4.9.1 TCML_CC

表 A.76 为 TCML_CC 的结构体定义。

**表 A.76 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<br>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的时候的返回值 |

#### A.4.9.2 TCML_CCA

表 A.77 为 TCML_CCA 的结构体定义。

**表 A.77 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

表 A.78 为 TCML_ALG 的结构体定义。

表 A.78 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<br>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的时候的返回值 |

### A.4.9.4 TCML_HANDLE

表 A.79 为 TCML_HANDLE 的结构体定义。

表 A.79 TCML_HANDLE 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| Count | UINT32 | 列表里句柄的个数<br>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的时候的返回值 |

### A.4.9.5 TCML_DIGEST

表 A.80 为 TCML_DIGEST 的结构体定义。

表 A.80 TCML_DIGEST 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| count {2:} | UINT32 | 列表里摘要的个数，最小为 2 或者 TCM2_PolicyOR() |
| 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的时候的返回值 |

### A.4.9.6 TCML_DIGEST_VALUES

表 A.81 为 TCML_DIGEST_VALUES 的结构体定义。

表 A.81 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

表 A.82 为 TCM2B_DIGEST_VALUES 的结构体定义。

**表 A.82 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

表 A.83 为 TCML_PCR_SELCETION 的结构体定义。

**表 A.83 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

表 A.84 为 TCML_ALG_PROPERTY 的结构体定义。

**表 A.84 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_PRPERTY

表 A.85 为 TCML_TAGGED_TCM2_PRPERTY 的结构体定义。

**表 A.85 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

表 A.86 为 TCML_TAGGED_PCR_PROPERTY 的结构体定义。

表 A.86　TCML_TAGGED_PCR_PROPERTY 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| Count | UINT32 | 属性个数<br>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

表 A.87 为 TCML_ECC_CURVE 的结构体定义。

表 A.87　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

表 A.88 为 TCMU_CAPABILITIES 的定义。

表 A.88　TCMU_CAPABILITIES 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_CC | TCM2_CAP_COMMANDS | |
| ppCommands | TCML_CC | TCM2_CAP_PP_COMMANDS | |
| 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

表 A.89 为 TCMS_CAPABILITY_DATA 的结构体定义。

表 A.89 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

表 A.90 为 TCMS_CLOCK_INFO 结构体定义。

表 A.90 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

表 A.91 为 TCMS_TIME_INFO 结构体定义,该结构用于 TCM2 GetTime()。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。如果需要,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.91 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

表 A.92 为 TCMS_TIME_ATTEST_INFO 结构体定义,当执行 TCM2 GetTime 时使用此结构。

表 A.92 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

表 A.93 为 TCMS_CERTIFY_INFO 结构体定义，这是 TCM2_Certify()的证明数据。

**表 A.93 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

表 A.94 为 TCMS_QUOTE_INFO 结构体定义，是 TCM2 Quote()的证明数据。

**表 A.94 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

表 A.95 为 TCMS_CREATION_INFO 结构体定义，是 TCM2_CertifyCreation()的证明数据。

**表 A.95 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

表 A.96 为 TCMI_ST_ATTEST 结构体定义。

**表 A.96 TCMI_ST_ATTEST 结构体定义**

| 值 | 描述 |
|---|---|
| TCM_ST_ATTEST_CERTIFY | TCM2_Certify()产生 |
| TCM_ST_ATTEST_QUOTE | TCM2_Quote()产生 |
| TCM_ST_ATTEST_COMMAND_AUDIT | TCM2_GetCommandAuditDigest()产生 |
| TCM_ST_ATTEST_TIME | TCM2_GetTime()产生 |
| TCM_ST_ATTEST_CREATION | TCM2_CertifyCreation()产生 |

#### A.4.12.6 TCMU_ATTEST

表 A.97 为 TCMU_ATTEST 结构体定义。

**表 A.97 TCMU_ATTEST 结构体定义**

| 参数 | 类型 | 值 |
|---|---|---|
| certify | TCMS_CERTIFY_INFO | TCM_ST_ATTEST_CERTIFY |
| creation | TCMS_CREATION_INFO | TCM_ST_ATTEST_CREATION |
| quote | TCMS_QUOTE_INFO | TCM_ST_ATTEST_QUOTE |
| commandAudit | TCMS_COMMAND_AUDIT_INFO | TCM_ST_ATTEST_COMMAND_AUDIT |
| sessionAudit | TCMS_SESSION_AUDIT_INFO | TCM_ST_ATTEST_SESSION_AUDIT |
| time | TCMS_TIME_ATTEST_INFO | TCM_ST_ATTEST_TIME |
| nvDigest | TCMS_NV_DIGEST_CERTIFY_INFO | TCM_ST_ATTEST_NV_DIGEST |

#### A.4.12.7 TCMS_ATTEST

表 A.98 为 TCMS_ATTEST 结构体定义。

**表 A.98 TCMS_ATTEST 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| magic | TCM_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

表 A.99 为 TCM2B_ATTEST 结构体定义。

此缓冲区可包含签名结构。证明数据是结构的签名部分。

**表 A.99 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

表 A.100 为 TCMS_AUTH_COMMAND 结构体定义，是用于命令会话区域中每个授权的结构。

**表 A.100 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

表 A.101 为 TCMS_AUTH_RESPONSE 结构体定义，是响应会话区域中每个授权的结构。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.101 TCMS_AUTH_RESPONSE 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| nonce | TCM2B_NONCE | 会话 nonce，可是空缓冲区 |
| 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

表 A.102 为 TCMI_! ALG.S_KEY_BITS 的类型定义，密钥大小以位(bit)表示。

**表 A.102 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

表 A.103 为 TCMU_SYM_KEY_BITS 联合体定义。

表 A.103　TCMU_SYM_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<br>注意:不允许 TCM_ALG_NULL |
| null | | TCM2_ALG_NULL | |

### A.5.1.3　TCMU_SYM_MODE

表 A.104 为 TCMU_SYM_MODE 联合体定义。

表 A.104　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

表 A.105 为 TCMT_SYM_DEF_OBJECT 结构体定义。

表 A.105　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

表 A.106 为 TCMU_SYM_DETAILS 联合体定义。

**表 A.106　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

表 A.107 为 TCM2B_SYM_KEY 结构体定义。

此结构用于在非对称对象的敏感区域中保存对称密钥。

密钥中的位数以公共区域中的密钥位数为单位。当密钥位数不是 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.107　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(八位字节)；可为零 |
| buffer [size] {:MAX_SYM_KEY_BYTES} | BYTE | 密钥 |

### A.5.1.7　TCM2B_DERIVE

表 A.108 为 TCM2B_DERIVE 结构体定义。

**表 A.108　TCM2B_DERIVE 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| size | UINT16 | |
| 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

表 A.109 为 TCM2B_SENSITIVE_CREATE 结构体定义。

此结构在大小合适的缓冲区中包含敏感的创建数据。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.109　TCM2B_SENSITIVE_CREATE 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| size= | UINT16 | 敏感字节的大小(不能为零)<br>“=”表示需要 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必须与 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

表 A.110 为 TCM2B_SENSITIVE_DATA 结构体定义。

缓冲区保存数据对象的秘密数据。它可以保存多达 128 个八位字节的数据，MAX_SYM_DATA 值应为 128。

**表 A.110 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 TCM2_ECC_PARAMETER

表 A.111 为 TCM2B_ECC_PARAMETER 结构体定义。

此大小的缓冲区包含 TCM 支持的最大 ECC 参数(坐标)。

**表 A.111 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.5.2.2 TCMS_ECC_POINT

表 A.112 为 TCMS_ECC_POINT 结构体定义。

这个结构包含两个 ecc 坐标，它们一起构成一个 ecc 点。

**表 A.112 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.5.2.3 TCM2B_ECC_POINT

表 A.113 为 TCM2B_ECC_POINT 结构体定义。

此大小的缓冲区包含 TCM 支持的最大 ECC 参数(坐标)。

表 A.113 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

表 A.114 为 TCMU_SIGNATURE 结构体定义。

表 A.114 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(需要支持) |
| 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

表 A.115 为 TCMT_SIGNATURE 结构体定义。

对称或非对称签名时的基本算法灵活结构。sigAlg 参数表示用于签名的算法。

表 A.115 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 密钥/秘密交换

#### A.5.4.1 TCMU_ENCRYPTED_SECRET

表 A.116 为 TCMU_ENCRYPTED_SECRET 联合体定义。

表 A.116 TCMU_ENCRYPTED_SECRET 联合体定义

| 参数 | 类型 | 选择器 | 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

表 A.117 为 TCM2B_ENCRYPTED_SECRET 结构体定义。

**表 A.117 TCM2B_ENCRYPTED_SECRET 结构体定义**

| 参数 | 类型 | 描述 |
|---|---|---|
| size | UINT16 | 秘密数据大小 |
| secret[size] {:sizeof(TCMU_ENCRYPTED_SECRET)} | BYTE | 秘密数据 |

### A.5.5 公钥数据结构

#### A.5.5.1 TCMI_ALG_PUBLIC

表 A.118 为 TCMI_ALG_PUBLIC 结构体定义。

**表 A.118 TCMI_ALG_PUBLIC 结构体定义**

| 值 | 描述 |
|---|---|
| TCM_ALG_! ALG.o | 所有对象类型 |
| #TCM_RC_TYPE | 不支持公共类型时的响应代码 |

#### A.5.5.2 TCMU_PUBLIC_ID

表 A.119 为 TCMU_PUBLIC_ID 结构体定义，是 TCMT_PUBLIC 的字段中允许的所有值的并集。

**表 A.119 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.5.5.3 TCMS_ASYM_PARMS

表 A.120 为 TCMS_ASYM_PARMS 结构体定义，此结构包含非对称密钥的公共区域参数。

**表 A.120 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。<br>对于具有加密属性的密钥，有效的密钥交换协议 |

### A.5.5.4 TCMS_ECC_PARMS

表 A.121 为 TCMS_ECC_PARMS 结构体定义，该结构包含 SM2 和 ECC 非对称密码算法的参数。

**表 A.121 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。<br>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。<br>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。<br>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。<br>如果与 curveID 关联的 kdf 参数不是 TCM2_ALG_NULL，则此参数必须为 NULL |

### A.5.5.5 TCMU_PUBLIC_PARMS

表 A.122 为 TCMU_PUBLIC_PARMS 结构体定义，定义了可能包含在密钥的公共部分中的参数定义结构。

**表 A.122 TCMU_PUBLIC_PARMS 结构体定义**

| 参数 | 类型 | 选择符 | 描述[a] |
|---|---|---|---|
| keyedHashDetail | TCMS_KEYEDHASH_PARMS | TCM2_ALG_KEYEDHASH | 签名\|解密\|全不[b] |
| symDetail | TCMS_SYMCIPHER_PARMS | TCM2_ALG_SYMCIPHER | 签名\|解密\|全不[b] |
| eccDetail | TCMS_ECC_PARMS | TCM2_ALG_ECC | 解密+签名[b] |
| asymDetail | TCMS_ASYM_PARMS | | ECC 密钥的通用方案结构 |

[a] 描述的列指明 TPMA_OBJECT.decrypt 或 PMA_OBJECT.sign 哪个可以设置。

[b] “+”表示两个都可以设置，但必须设置一个。“|”表示可选设置。

### A.5.5.6 TCMT_PUBLIC

表 A.123 为 TCMT_PUBLIC 结构体定义，使用算法名称将其与此结构的摘要连接起来。

**表 A.123 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 |

表 A.123 TCMT_PUBLIC 结构体定义（续）

| 参数 | 类型 | 描述 |
|---|---|---|
| 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.5.5.7 TCM2B_PUBLIC

表 A.124 为 TCM2B_PUBLIC 结构体定义，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。

表 A.124 TCM2B_PUBLIC 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| size= | UINT16 | publicArea 大小<br>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）<br>“=”表示需要 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必须与 size 相同 |
| publicArea | +TCMT_PUBLIC | 公共结构数据<br>注意，“+”表示调用方可指定允许对 nameAlg 使用 TCM2_ALG_NULL |

### A.5.5.8 TCM2B_TEMPLATE

表 A.125 为 TCM2B_TEMPLATE 结构体定义，此大小的缓冲区用于为 TCM2_CreateLoaded()嵌入 TCMT_TEMPLATE 模板。

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.125 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.5.6 私钥数据结构

### A.5.6.1 TCMU_SENSITIVE_COMPOSITE

表 A.126 为 TCMU_SENSITIVE_COMPOSITE 结构体定义。

表 A.126 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.5.6.2 TCMT_SENSITIVE

表 A.127 为 TCMT_SENSITIVE 结构体定义。授权值不应大于由对象的算法名称生成的摘要的大小。种子值应为对象算法名称生成的摘要的大小。

表 A.127 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。<br>这应与相关公共区域的类型参数相同 |
| authValue | TCM2B_AUTH | 用户授权数据。<br>授权值可是零长度字符串 |
| 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.5.6.3 TCM2B_SENSITIVE

表 A.128 为 TCM2B_SENSITIVE 结构体定义。该结构用作 TCM2_LoadExternal()中的参数。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.128 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.5.6.4 TCM2B_PRIVATE

表 A.129 为 TCM2B_PRIVATE 结构体定义，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。

表 A.129 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.5.7 标识对象

#### A.5.7.1 TCMS_ID_OBJECT

表 A.130 为 TCMS_ID_OBJECT 结构体定义。

表 A.130 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。<br>所有的加密，包括大小字段 |

#### A.5.7.2 TCM2B_ID_OBJECT

表 A.131 为 TCM2B_ID_OBJECT 结构体定义。

表 A.131 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.6 NV 存储结构

### A.6.1 TCM2_NV_INDEX

表 A.132 为 TCM2_NV_INDEX 结构体定义，TCM_NV_INDEX 用于引用 NV 内存中定义的位置。

表 A.132 TCM2_NV_INDEX 结构体定义

| 位 | 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.6.2 TCMA_NV

表 A.133 为 TCMA_NV 结构体定义。

此结构允许 TCM 跟踪数据和操作 NV 索引的权限。

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.133　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。<br>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。<br>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。<br>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):策略会话提供更改需要用户角色的索引内容的授权。<br>CLEAR(0):更改需要用户角色的索引内容的授权不能与策略会话一起提供 |
| 4 | TCMA_NV_COUNTER | SET(0):索引包含一个 8 位字节的值,该值将用作计数器,并且只能用 TCM2_NV_Increment()修改<br>CLEAR(0):索引不是计数器 |
| 5 | TCMA_NV_BITS | SET(1):索引包含一个 8 字节的值,用作位字段,只能用 TPM2_NV_SetBits()修改。<br>CLEAR(0):索引不是位字段 |
| 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 TPM2_NV_ Extend 修改索引。extend 将使用索引的 nameAlg。<br>SET(0):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。<br>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。<br>CLEAR(0):可写入索引 |

表 A.133 TCMA_NV 结构体定义（续）

| 位 | 名称 | 描述 |
| --- | --- | --- |
| 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。<br>CLEAR(0)：允许部分写入。如果索引的.dataSize 大于实现的 NV_MAX_BUFFER_SIZE，则需要此设置 |
| 13 | TCMA_NV_WRITEDEFINE | SET(1)：TCM2_NV_WriteLock()可用于防止进一步写入此位置。<br>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。<br>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。<br>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。<br>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。<br>CLEAR(0)：无法使用所有者授权读取索引数据 |
| 18 | TCMA_NV_AUTHREAD | SET(1)：如果提供了索引授权值，则可读取索引数据。<br>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。<br>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。<br>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 |
| 26 | TCMA_NV_ORDERLY | SET(1)：仅当 TCM 执行断电(TCM2_shutdown())时，才需要保存 NV 索引状态。<br>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。<br>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。<br>CLEAR(0)：如果提供了适当的授权，则允许读取索引 |

表 A.133 TCMA_NV 结构体定义（续）

| 位 | 名称 | 描述 |
|---|---|---|
| 29 | TCMA_NV_WRITTEN | SET(1):索引已写入。<br>CLEAR(0):索引未写入 |
| 30 | TCMA_NV_PLATFORMCRE-ATE | SET(1):此索引可能未经平台授权定义,但未经所有者授权。<br>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。<br>CLEAR(0):TCM2_NV_ReadLock()对该索引没有影响 |

### A.6.3 TCMS_NV_PUBLIC

表 A.134 为 TCMS_NV_PUBLIC 结构体定义,此结构描述一个 NV 索引。

表 A.134 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.6.4 TCM2B_NV_PUBLIC

表 A.135 为 TCM2_BNV_NV_PUBLIC 结构体定义,在 TCM 接口上发送 TCMS_NV_PUBLIC 时使用此结构。

表 A.135 TCM2_BNV_NV_PUBLIC 结构体定义

| 名称 | 类型 | 描述 |
|---|---|---|
| size= | UINT16 | nvPublic 大小<br>“=”表示需要 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必须与 size 相同 |
| nvPublic | TCMS_NV_PUBLIC | 公共区域 |

## A.7 上下文数据

### A.7.1 TCM2B_CONTEXT_SENSITIVE

表 A.136 为 TCM2B_CONTEXT_SENSITIVE 结构体定义。

**表 A.136 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.7.2 TCMS_CONTEXT_DATA

表 A.137 为 TCMS_CONTEXT_DATA 结构体定义。

**表 A.137 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.7.3 TCM2B_CONTEXT_DATA

表 A.138 为 TCM2B_CONTEXT_DATA 结构体定义。

**表 A.138 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.7.4 TCMS_CONTEXT

表 A.139 为 TCMS_CONTEXT 结构体定义。

**表 A.139 TCMS_CONTEXT 结构体定义**

| 名称 | 类型 | 描述 |
|---|---|---|
| sequence | UINT64 | 上下文的序号<br>**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.7.5 TCMS_CONTEXT 值

表 A.140 为 TCMS_CONTEXT 值定义。

**表 A.140 TCMS_CONTEXT 值定义**

| 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.7.6 TCMT_TK_CREATION

表 A.141 为 TCMT_TK_CREATION 结构体定义。

**表 A.141 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 | 可是使用等级证明值 HMAC 产生 |

### A.7.7 TCMT_TK_VERIFIED

表 A.142 为 TCMT_TK_VERIFIED 结构体定义。

**表 A.142 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 | 可是使用等级证明值 HMAC 产生 |

### A.7.8 TCMT_TK_AUTH

表 A.143 为 TCMT_TK_AUTH 结构体定义。

表 A.143　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 | 可是使用等级证明值 HMAC 产生 |

### A.7.9　TCMT_TK_HASHCHECK

表 A.144 为 TCMT_TK_HASHCHECK 结构体定义。

表 A.144　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 | 可是使用等级证明值 HMAC 产生 |

### A.7.10　TCMI_ALG_PUBLIC

表 A.145 为 TCM1_ALG_PUBLIC 类型定义。

表 A.145　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.7.11　TCMT_PUBLIC

表 A.146 为 TCMT_PUBLIC 结构体定义。

表 A.146　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<br>**注：**<br>“+”表明实例可有 TCMT_PUBLIC<br>一个“+”表明 nameAlg 可为 TCM2_ALG_NULL |

表 A.146 TCMT_PUBLIC 结构体定义（续）

| 参数 | 类型 | 描述 |
|---|---|---|
| objectAttributes | TCMA_OBJECT | 与类型的类型，确定了这个对象的操作 |
| authPolicy | TCM2B_DIGEST | 使用密钥的可选策略<br>使用对象的 nameAlg 来计算策略<br>**注**：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<br>一个非对称密钥，这个应是公钥 |

### A.7.12 TCM2B_PUBLIC

表 A.147 为 TCM2B_PUBLIC 结构体定义。

表 A.147 TCM2B_PUBLIC 结构体定义

| 参数 | 类型 | 描述 |
|---|---|---|
| Size= | UINT16 | publicArea 大小<br>“=”表示需要 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必须与 size 相同 |
| publicArea | +TCMT_PUBLIC | 公共区域<br>“+”表明调用者可指定 nameAlg 使用 TCM2_ALG_NULL |

中华人民共和国密码
行业标准
**可信计算　可信密码模块接口规范**
GM/T 0012—2020
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)68533533　发行中心:(010)51780238
读者服务部:(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230　1/16　印张 8.5　字数 258 千字
2021年5月第一版　2021年5月第一次印刷
*
书号:155066·2-35852　定价 106.00 元



GM/T 0012-2020

# 可信计算　可信密码模块接口规范

中国标准出版社