ATECC608A-TNGTLS

ATECC608A-TNGTLS CryptoAuthentication™数据手册

简介

ATECC608A-TNGTLS 是 ATECC608A 的预配置型号。Trust&GO 安全元件属于 Microchip 通用配置安全器件系列。器件配置旨在使安全元件适用于 IoT 市场中的一些最常见用例，同时最大程度地减少与安全器件相关的学习和入门准备。

本数据手册提供了 ATECC608A-TNGTLS 特有的槽和密钥配置信息。该信息明确定义了每个数据区域槽的访问策略。其中仅包含相关命令和 I/O 工作信息。此外，应用部分介绍了可帮助开发应用程序的 Microchip 软硬件工具及其下载链接。

特性

- 完全指定的配置区域
- 可一次性更改 I^2C 地址的 I^2C 接口
- 一个永久的主 P-256 椭圆曲线加密技术（Elliptic Curve Cryptography，ECC）私钥，在制造阶段固定
- 一个用于密钥认证的内部签名私钥
- 三个可由用户重新生成的辅助 P-256 ECC 私钥
- 来自签名人证书的签名人公钥
- IO 保护密钥槽，用于保护 I^2C 通信
- 可与 AES 密钥和命令搭配使用的 ECDH/KDF 密钥槽
- X.509 压缩证书存储
- 采用 8 焊盘 UDFN 和 8 引脚 SOIC 封装（2k 或 100 件）。

应用

- 安全 IoT TLS 1.2 和 1.3 连接

© 2020 Microchip Technology Inc.

目录

简介 1

特性 1

应用 1

1. 引脚配置和引脚分配 4

2. EEPROM 存储器和数据区域访问策略 5
   - 2.1. ATECC608A-TNGTLS 配置区域 5
   - 2.2. 数据区域和访问策略 8
   - 2.3. ATECC608A-TNGTLS EEPROM 可一次性编程（OTP）区域 14

3. 静态 RAM（SRAM）存储器 16
   - 3.1. TempKey 16
   - 3.2. 报文摘要缓冲区 16
   - 3.3. 备用密钥缓冲区 17
   - 3.4. SHA 上下文缓冲区 17

4. 一般命令信息 18
   - 4.1. I/O 事务 18
   - 4.2. 命令数据包 18
   - 4.3. 状态/错误代码 18
   - 4.4. 地址编码 19
   - 4.5. 密钥、签名和证书的格式 21

5. 器件命令 25
   - 5.1. 常规器件命令 25
   - 5.2. 非对称加密命令 33
   - 5.3. 对称加密命令 42

6. 应用信息 57
   - 6.1. 用例 57
   - 6.2. 开发工具 57
   - 6.3. TrustFLEX 与 Trust&GO 58

7. I^2C 接口 60
   - 7.1. I/O 条件 60
   - 7.2. 到 ATECC608A-TNGTLS 的 I^2C 传输 ATECC608A-TNGTLS 61
   - 7.3. 休眠序列 63
   - 7.4. 空闲序列 63
   - 7.5. 来自 ATECC608A-TNGTLS 的 I^2C 传输 ATECC608A-TNGTLS 63

8. 电气特性 65
   - 8.1. 绝对最大值 65
   - 8.2. 可靠性 65
   - 8.3. 交流参数：所有 I/O 接口 65

8.4. 直流参数：所有 I/O 接口 67
8.5. V_{IH} 和 V_{IL} 规范 68

9. 封装图 69
9.1. 封装标识信息 69
9.2. 8 焊盘 UDFN 69
9.3. 8 引脚 SOIC 72

10. 版本历史 75

Microchip 网站 76

产品变更通知服务 76

客户支持 76

产品标识体系 77

Microchip 器件代码保护功能 78

法律声明 78

商标 78

质量管理体系 79

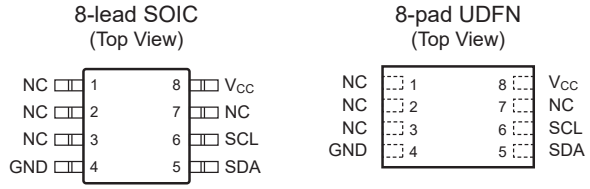
全球销售及服务网点 80

1. 引脚配置和引脚分配

表 1-1. 引脚配置

引脚	功能
NC	无连接
GND	地
SDA	I^2C 串行数据
SCL	I^2C 串行时钟输入
V_{CC}	电源

图 1-1. UDFN 和 SOIC 引脚分配



注： UDFN 背面的焊盘应连接至 GND。

2. EEPROM 存储器和数据区域访问策略

EEPROM 存储器共有 1,400 字节，分为以下几个区域：

表 2-1. ATECC608A-TNGTLS EEPROM 区域

区域	说明	命名法
配置	128 字节（1,024 位）EEPROM 区域，其中包含： • 器件配置 • 槽访问策略信息 • 计数器值 • 器件序列号 • 锁定信息 已设置 LockConfig 字节。无法直接向此区域写入任何内容。始终可以读取该区域。	Config[a:b] = 配置区域中的字段内的字节范围。
数据	1,208 字节（9.7 Kb）区域，分为 16 个通用只读或读/写存储器槽。槽的划分如下： • Slot 0-7 包含 36 个字节 • Slot 8 包含 416 个字节 • Slot 9-15 包含 72 个字节 配置区域字节定义的访问策略信息决定了每个槽的访问方式。ATECC608A-TNGTLS 器件中每个数据槽的访问策略均已设置并且配置区域定义的槽访问策略全面生效。一些槽可以读写，而另一些槽无法读写，具体取决于槽访问策略。	Slot[YY] = 存储在数据区域的 Slot YY 中的全部内容。
可一次性编程（OTP）	64 字节（512 位）区域，分为两个块，每个块 32 个字节。对于 ATECC608A-TNGTLS，该区域已预装载有预定义的值。该区域无法修改，但可随时读取。有关更多信息，请参见 2.3 ATECC608A-TNGTLS EEPROM 可一次性编程（OTP）区域。	OTP[bb] = OTP 区域内的一个字节，而 OTP[aa:bb]表示字节范围。

表 2-2. 文档术语

本文档中讨论的术语具有以下含义：

术语	含义
块	特定存储区域的单个 256 位（32 字节）区域。工业 SHA-256 文档还使用术语“块”来表示报文输入的 512 位片段。在本文档内，仅当描述哈希输入报文时使用此约定。
KeyID	KeyID 相当于指定为保存密钥值的槽的编号。Key1（有时称为 key[1]）存储在 Slot[1]中，依此类推。当全部 16 个槽均可保存密钥时，配置为允许明文读取的槽通常不会被加密命令用作私钥或机密信息密钥。
mode[b]	表示参数模式的 bit b。
SRAM	包含输入和输出缓冲区以及状态存储单元。用户不能直接访问此存储器。请参见 3. 静态 RAM（SRAM）存储器。
字	从块读取或写入块的单个 4 字节数据字。字是数据访问的最小单位。

2.1 ATECC608A-TNGTLS 配置区域

配置区域中的 128 个字节包含制造标识数据、通用器件和系统的配置信息以及数据区域中槽的访问策略控制值。它分为 4 个块，每个块 32 字节。这些字节的值始终可使用 read 命令获得。

对于 ATECC608A-TNGTLS 而言，这些值中的大多数是预先配置的固定值。 2.1.1 可修改的配置区域字节将讨论锁定后可修改的字节。

此区域字节的配置如下表所示。

表 2-3. ATECC608A-TNGTLS 配置区域设置

字节	名称	配置值 [LSB MSB]	说明
[0:3]	SN[0:3]	01 23 xx xx	序列号值部分。
[4:7]	RevNum	00 00 60 02	器件版本号。
[8:12]	SN[4:8]	xx xx xx xx 01	序列号值部分。
[13]	AES_Enable	01	使能 AES 操作。
[14]	I2C_Enable	xx	**b[7:1]** 由 Microchip 编程，将因器件而异。 **b[0]** 1——对于 I²C 模式器件
[15]	保留	00	由 Microchip 设置，将始终为 0x00。
[16]	I2C_Address	6A	默认 7 位 I²C 地址为 0x35。
[17]	保留	00	保留。必须为 0。
[18]	CountMatch	00	禁止计数器匹配功能
[19]	ChipMode	01	**b[7:3]** 0x00 时钟分频器模式为高速 **b[2]** 0 看门狗时间设置为 1.3s **b[1]** 0 I/O 使用固定参考模式 **b[0]** 1 使能备用 I²C 地址模式
[20:51]	SlotConfig	请参见 2.2.4	两个字节的访问和使用权限区域，可控制数据区域的各个槽。
[52:59]	Counter[0]	FF FF FF FF 00 00 00 00	单调计数器 0 未关联到任何密钥，但可以根据需要用作系统计数器。
[60:67]	Counter[1]	FF FF FF FF 00 00 00 00	单调计数器 1 未关联到任何密钥，但可以根据需要用作系统计数器。
[68]	UseLock	00	禁止使用锁定密钥。
[69]	VolatileKey 权限	00	禁止易失性密钥权限。
[70:71]	Secure Boot	xx xx	未使用安全引导。
[72]	KdfIvLoc	00	由于 ChipOptions.KDFPROT 在此配置中不会强制加密，因此无效。
[73:74]	KdfIvStr	69 76	由于 ChipOptions.KDFPROT 在此配置中不会强制加密，因此无效。
[75:83]	保留	0	必须为 0。
[84]	UserExtra	00	数据区域锁定后可通过 UpdateExtra 命令修改的 1 字节值。仅当其值为 0 时才能通过 UpdateExtra 写入。

.....（续）

字节	名称	配置值 [LSB MSB]	说明
[85]	UserExtraAdd	00	如果该字节的值!= 0x00，则该字节将为器件的 I²C 地址。如果值为 0x00，则可以通过 UpdateExtra 命令写入该值。
[86]	LockValue	00	数据区域已锁定，因此该值将为 0x00。
[87]	LockConfig	00	配置区域已锁定，因此该值将为 0x00。
[88:89]	SlotLocked	FF FF	对于可锁定的槽，表示给定槽的锁定状态。0 = 锁定 1 = 未锁定 对于 ATECC608A-TNGTLS，以下各槽可能被锁定：Slot 2-4、6 和 Slot 8。对于所有其他槽，该值将为 1
[90:91]	ChipOptions	0E 60	**b[15:12]** 0x6 IO 保护密钥设置为 Slot 6 **b[11:10]** 0b00 允许以明文形式输出 KDF 函数，但可以进行加密 **b[9:8]** 0b00 允许以明文形式输出 ECDH 主机密信息，但可以进行加密 **b[7:4]** 必须为 0 **b[3]** 1 = 由于运行状况测试失败而导致命令失败后，运行状况测试失败位将清零。如果失败现象是暂时的，则在第二次运行时，命令可能会通过。 **b[2]** 1 使能 KDF AES 模式 **b[1]** 1 使能 IO 保护密钥 **b[0]** 0 唤醒时禁止上电自检
[92:95]	X509format	00 00 00 00	禁止/忽略证书格式。
[96:127]	KeyConfig	请参见 2.2.4	两个字节的附加访问和使用权限区域，可控制数据区域的各个槽。

2.1.1 可修改的配置区域字节

由于配置区域已锁定，因此不能直接写入配置区域中的任何字节。但是，仍可使用其他命令来修改某些字节。

SlotLocked 位

对于 ATECC608A-TNGTLS，以下各槽可能被锁定：Slot 2-4、6 和 Slot 8。这些槽可通过 Lock 命令的 SlotLock 模式锁定。每个使能了此功能的槽只能被单独锁定一次。一旦槽被锁定，将无法对其进行修改或解锁，但仍可根据为该槽定义的访问策略进行使用。

I²C 地址重新定义

此器件配置已创建，以便可重新定义一次 I²C 地址。UpdateExtra 命令可用于将配置区域的字节 85 重新写入新 I²C 地址。当该字节设置为非零值时，器件配置将字节 85 用作其 I²C 地址，而非使用默认地址。重新写入该字节后，器件必须先掉电或置于休眠模式，之后此更改才会生效。

重要： 如果无需更改 I²C 地址，则应使用默认 I²C 地址写入该存储单元。

UserExtra 字节
UserExtra 字节可用于任何所需用途。该字节只能用 UpdateExtra 命令更新一次。

Counter[0,1]
尽管该器件未使用计数器，但不会将其禁止。如有必要，系统可使用单调计数器。请注意，计数器被初始化为零，并且可以计数到最大值 2,097,151。计数器值可递增或通过使用 Counter 命令读取。该计数器的使用方式严格遵循系统要求，与器件上的其他任何事物无关。

2.2 数据区域和访问策略

以下各节介绍了与每个槽关联的详细访问策略信息。实际访问策略信息存储在 EEPROM 配置区域的槽和密钥配置段中。每个数据区域槽都具有与其关联的 2 个槽配置字节和 2 个密钥配置字节。这 4 个字节一起为每个槽创建“访问策略”。存储在槽中的数据的实际类型由该槽的访问策略确定。

2.2.1 数据区域数据类型

以下各节提供了有关可以存储在 ATECC608A-TNGTLS 数据槽中的各个数据类型的详细信息。

2.2.1.1 私钥

ECC 私钥是 ECC 安全性的基本构件。这些密钥是私密的，并且对于每个器件都是惟一的，始终无法读取。ECC 私钥由安全元件的 TRNG 随机生成，并安全地保存在配置为 ECC 私钥的槽中。

主私钥
这是主身份验证密钥。该私钥是永久性的，不能更改。每个器件都有自己惟一的私钥。

由以下两种主椭圆曲线函数生成该密钥：

- 用于身份验证的 ECDSA 签名
- 用于密钥协议的 ECDH。如果需要对 ECDH 输出进行加密，则需要先设置 IO 保护密钥。有关设置的详细信息，请参见 2.2.1.4 I/O 保护密钥。

该私钥是生成相应公钥和 X.509 证书的基础。

辅助私钥
提供可供将来使用（例如附加服务身份验证）的附加私钥。

由以下两种主椭圆曲线函数生成此类密钥：

- 用于身份验证的 ECDSA 签名。
- 用于密钥协议的 ECDH。如果需要对 ECDH 输出进行加密，则需要先设置 IO 保护密钥。有关设置的详细信息，请参见 2.2.1.4 I/O 保护密钥。
- GenKey，用于使用在内部生成的新随机私钥覆盖槽。

尽管主密钥和证书是永久性的，但其他密钥可以用内部生成的新私钥覆盖（GenKey 命令模式 = 0x04），以使能密钥删除、密钥轮换和远程配置。这些密钥也是槽可锁定的（KeyConfig.Lockable 位设置为零），这意味着 Lock 命令可用于槽锁定模式，以使当前密钥成为永久密钥，并防止 GenKey 命令更改该密钥。更改密钥时，需要通过密钥认证来确保新密钥在 ATECC608A-TNGTLS 器件中得到妥善保护，之后才能将其视为可信密钥。

密钥认证
Slot 1 中的私钥配置为仅内部签名密钥，这意味着，它只能签名由 GenKey 或 GenDig 命令在内部生成的报文，并且不能用于签名任意外部报文。通过此功能，可将内部签名密钥用于向任何已知（并信任）内部签名公钥的系统证明器件中的密钥及其配置/状态。

2.2.1.2 证书动态存储

ATECC608A-TNGTLS 存储围绕密钥进行。X.509 证书的大小通常比 ATECC608A-TNGTLS 槽大，因此使用压缩格式。该技术更应被称为部分证书，因为它在器件上存储动态证书信息，并施加了一些限制。动态信息是可以随器件更改的证书内容（例如，公钥和有效日期等）。固件应具有证书定义（来自 CryptoAuthLib 的 atcacert_def_t）与完整 X.

509 证书的模板，证书中包含静态信息（对于所有证书均相同的数据）以及有关如何利用压缩证书中的动态信息重新编译完整证书的说明。

以下应用笔记记录了压缩证书格式：ATECC Compressed Certificate Definition。

CryptoAuthLib 库还包含用于处理压缩证书的 atcacert 模块。

器件证书
器件证书包含与实际最终器件关联的信息。对于 ATECC608A-TNGTLS，器件证书存储在 Slot 10 中。

签名人证书
签名人证书包含与用于签名器件证书的签名人相关的信息。对于 ATECC608A-TNGTLS，签名人证书存储在 Slot 12 中。此外，还需要添加签名人公钥来构成完整的签名人证书。

签名人公钥
签名人公钥用于验证签名人和与签名人压缩证书关联的信息。对于 ATECC608A-TNGTLS，签名人公钥存储在 Slot 11 中。

下表列出了与 ATECC608A-TNGTLS 中的证书关联的所有槽：

槽	说明
0	主私钥。在模式 = 0x00 时，可使用 GenKey 命令随时生成公钥。
10	器件证书。以压缩格式存储在此处。请参见 4.5.3 证书存储。
11	签名人公钥。请参见 4.5.1.1 公钥格式。
12	签名人证书。以压缩格式存储。请参见 4.5.3 证书存储。

这些证书也是永久性的，用于存储这些证书的槽不能更改。

2.2.1.3 AES 密钥存储

ECDH 和 KDF 等命令输出对称密钥。上述命令还可将这些密钥保存到一个槽中以实现安全存储和使用。AES 密钥存储槽已设置为这些密钥的目标槽。多个密钥可以存储在一个槽中。

2.2.1.4 I/O 保护密钥

Verify、ECDH、和 KDF 命令还可选择使用 I/O 保护功能来加密某些参数并（通过 MAC）验证一些响应。这有助于防止物理 I^2C 总线上出现中间人攻击。但是，为了能够使用该功能，MCU 和 ATECC608A-TNGTLS 需要事先生成并保存惟一的 I/O 保护密钥，实际上是将 MCU 和 ATECC608A-TNGTLS 器件彼此配对。必须在首次引导时进行配对过程。

I/O 保护密钥生成：

1. MCU 使用随机命令来生成一个随机的 32 字节 I/O 保护密钥。
2. MCU 将 I/O 保护密钥保存在其内部闪存中。
3. MCU 将 I/O 保护密钥写入 I/O 保护密钥槽中。
4. MCU 槽锁定该槽以使 I/O 保护密钥具有永久性。

作为配对检查，MCU 可以使用 MAC 命令向 I/O 保护密钥发出质询，并验证闪存中存储的 I/O 保护密钥是否与 ATECC608A-TNGTLS 中的密钥匹配。

2.2.1.5 通用数据存储

为支持通用公共数据存储，已设置了许多槽。这些槽可用于存储任何允许公开访问的数据。这些槽始终可采用明文形式读取和写入。

2.2.2 槽配置术语

以下部分给出了一组用于说明配置选项的术语。

术语	说明
AES 密钥	槽可用作 AES 命令的密钥源。对于 ATECC608A-TNGTLS，AES 密钥的宽度为 128 位。
始终写入	可使用写命令以明文形式写入槽。
明文读取	槽被认为是公共的（非机密的），其内容可使用读命令以明文形式读取。
ECDH	椭圆曲线 Diffie Hellman。私钥可与 ECDH 命令搭配使用。
外部签名	私钥可用于签名外部（任意）报文。
内部签名	私钥可用于签名由 GenKey 或 GenDig 命令生成的内部报文。用于证明器件的内部密钥和配置。
可锁定	槽可在未来某个时刻进行锁定。一旦锁定，槽内容将无法更改（仅供读取/使用）。
不可读取	槽被认为是机密的，其内容不能使用读命令进行读取。私钥和对称机密信息应始终配置为“不可读取”。
不可写入	无法使用写命令更改槽。
永久	私钥是永久性的/不可更改。它在出厂配置期间由内部生成。
可更新	私钥稍后可由内部生成的新随机私钥覆盖。其初始值在出厂配置期间由内部生成。

2.2.3 ATECC608A-TNGTLS 槽配置汇总

ATECC608A-TNGTLS 有 16 个槽，可以针对不同的用例进行配置。下表对 ATECC608A-TNGTLS 的这些槽及其配置和建议用例进行了汇总：

槽	用例	说明	主配置
0	主私钥	主身份验证密钥。	永久、外部签名和 ECDH
1	内部签名私钥	只能用于认证器件的内部密钥和状态的私钥。无法用于为任意报文签名。	永久，内部签名
2	辅助私钥 1	用于其他用例的辅助私钥。	可更新，外部签名，ECDH，可锁定
3	辅助私钥 2	用于其他用例的辅助私钥。	可更新，外部签名，ECDH，可锁定
4	辅助私钥 3	用于其他用例的辅助私钥。	可更新，外部签名，ECDH，可锁定
5	保留		
6	IO 保护密钥	用于保护某些命令的 I^2C 总线通信（IO）的密钥。使用前需要进行设置。	不可读取，始终写入，可锁定
7	保留		
8	通用数据	通用公共数据存储（416 字节）。	明文读取，始终写入，可锁定
9	AES 密钥	ECDH 和 KDF 输出的中间密钥存储位置。	无读取，始终写入，AES 密钥
10	器件压缩证书	CryptoAuthentication™压缩格式的证书主公钥。	明文读取，不可写入
11	签名人公钥	对器件证书进行签名的 CA（签名人）的公钥。	明文读取，不可写入
12	签名人压缩证书	采用 CryptoAuthentication™压缩格式的器件证书的 CA（签名人）证书的证书。	明文读取，不可写入
13	保留		

.....（续）

槽	用例	说明	主配置
14	保留		
15	保留		

2.2.4 ATECC608A-TNGTLS 详细槽访问策略

详细槽配置
下表提供了关于每个槽密钥和槽配置信息以及可以使用该槽运行哪些命令和命令模式的更详细说明。

表 2-4. Slot 0 配置信息

槽	配置值	已使能功能的说明
0	密钥：0x0053	**主私钥** • 包含 P256 NIST ECC 私钥（KeyType = 0x4）(1) • 始终可以生成相应的公钥 • 需要随机 nonce
	槽：0x0085	• 槽为机密信息槽 • 可以对外部报文进行签名 • 可以搭配使用 ECDH 命令
	有效命令	• GenKey——公钥生成 • 签名——外部报文 • ECDH——创建共享机密信息

表 2-5. Slot 1 配置信息

槽	配置值	已使能功能的说明
1	密钥：0x0053	**内部签名私钥** • 包含 P256 NIST ECC 私钥（KeyType = 0x4）(1) • 始终可以生成相应的公钥 • 需要随机 nonce
	槽：0x0082	• 槽为机密信息槽 • 可以对 GenDig 或 GenKey 生成的内部报文进行签名 • 禁止 ECDH
	有效命令	• GenKey——公钥生成 • 签名——内部报文（由 GenDig 或 GenKey 生成）

表 2-6. Slot 2-4 的槽和密钥配置

槽	配置值	已使能功能的说明
2、3 或 4	密钥：　0x0073	**辅助私钥 1-3** • 包含 P256 NIST ECC 私钥（KeyType = 0x4）(1) • 始终可以生成相应的公钥 • 使用该密钥时，始终需要随机 nonce • 该槽可以单独锁定
	槽：　0x2085	• GenKey 可在锁定之前用于在该槽中生成新的 ECC 私钥 • 槽为机密信息槽 • 可以对外部报文进行签名 • 可以搭配使用 ECDH 命令
	有效命令	• GenKey——私钥重新生成 • GenKey——公钥生成 • 签名——外部报文 • ECDH——创建共享机密信息 • 锁定——SlotLock 模式

表 2-7. Slot 5 配置信息

槽	配置值	已使能功能的说明
5	na　na	**保留**

表 2-8. Slot 6 配置信息

槽	配置值	已使能功能的说明
6	密钥：　0x007C	**IO 保护密钥** • 可包含 SHA256 对称密钥或其他数据。（KeyType = 0x7）(1)。如果不使用 IO 保护密钥，则该槽可用于其他数据 • 使用该密钥时需要随机 nonce • 该槽可以单独锁定
	槽：　0x0F8F	• 数据可采用明文形式写入 • 该槽的内容为机密信息，无法读取 • 槽无法用于 CheckMac Copy 命令
	有效命令	• 以明文形式写入 Slot 6 • MAC 命令 • 锁定——SlotLock 模式

表 2-9. Slot 7 配置信息

槽	配置值	已使能功能的说明
7	na　na	保留

表 2-10. Slot 8 配置信息

槽	配置值	已使能功能的说明
8	密钥：0x003C	**通用数据** • 该槽专用于和通用数据配合使用（KeyType = 0x7）(1) • 槽可锁定
	槽：0x0F0F	• 允许对该槽进行明文写入和读取。 • 槽无法用于 CheckMac Copy 命令
	有效命令	• 写入——明文 • 读取——明文 • GenDig——数据源 • MAC 命令 • 锁定——SlotLock 模式

表 2-11. Slot 9 配置信息

槽	配置值	已使能功能的说明
9	密钥：0x001A	**AES 密钥** • 槽最多可存储 4 个 AES 128 位对称密钥（KeyType = 0x6）(1)
	槽：0x0F8F	• 允许对该槽进行明文写入 • 该槽为机密信息槽 • 槽无法用于 CheckMac Copy 命令
	有效命令	• 写入——明文 • AES——加密/解密（源密钥） • MAC 命令

表 2-12. Slot 10 配置信息

槽	配置值	已使能功能的说明
10	密钥：0x001C	**器件压缩证书** • 定义为存储其他数据的槽。（KeyType = 0x7）(1)
	槽：0x8F0F	• 数据无法覆盖 • 数据可采用明文形式读取
	有效命令	• 读取——明文 • GenDig——数据源 • MAC 命令

表 2-13. Slot 11 配置信息

槽	配置值	已使能功能的说明
11	密钥：0x0010	**签名人公钥** • 槽被定义用于 ECC 密钥（KeyType——0x4）(1) • ECC 密钥为公钥
	槽：0x8F0F	• 数据无法覆盖 • 数据可采用明文形式读取
	有效命令	• 读取——明文 • GenDig——数据源 • Verify 命令 • MAC 命令 • GenKey——公共摘要模式

表 2-14. Slot 12 配置信息

槽	配置值	已使能功能的说明
12	密钥：0x001C	**签名人压缩证书** • 定义为存储其他数据的槽。（KeyType = 0x7）(1)
	槽：0x8F0F	• 数据无法覆盖 • 数据可采用明文形式读取
	有效命令	• 读取——明文 • GenDig——数据源 • MAC 命令

表 2-15. Slot 13 配置信息

槽	配置值		已使能功能的说明
13	na	na	**保留**

表 2-16. Slot 14 配置信息

槽	配置值		已使能功能的说明
14	na	na	**保留**

表 2-17. Slot 15 配置信息

槽	配置值		已使能功能的说明
15	na	na	**保留**

注：

1. KeyType 由每个槽的密钥配置位[4:2]指定。

2.3 ATECC608A-TNGTLS EEPROM 可一次性编程（OTP）区域

64 字节（512 位）的 OTP 区域是 EEPROM 阵列的一部分，可用于只读存储。它分为 2 个块，每个块 32 字节。对于 ATECC608A-TNGTLS，OTP 区域出厂时已预先锁定，其中包含以下信息：

```
52 73 75 79 35 59 4A 68 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

写入 OTP 区域中的数据字节值始终可通过 4 字节或 32 字节读取方式进行读取，但始终不可修改。

3. 静态 RAM（SRAM）存储器

此器件还包括用于存储输入命令或输出结果、nonce、中间计算值、临时密钥和 SHA 上下文等内容的 SRAM 阵列。SRAM 的内容始终不能直接读取；仅供安全元件在内部使用。每当器件进入休眠模式或断电时，此寄存器的全部内容便无效。

3.1 TempKey

TempKey 是 SRAM 阵列中的主存储寄存器，可用于存储各种中间值。此寄存器的内容始终不能从器件读取（尽管器件本身可以在内部读取和使用该内容）。

TempKey 的长度为 64 字节。KDF 和 Nonce 命令能够写入该寄存器的前 32 个字节和后 32 个字节；而所有其他命令均只能修改 TempKey 的前（低）32 个字节。只有前 32 个字节有效或全部 64 个字节有效这两种情况。如果低 32 个字节无效，则器件不允许高 32 个字节有效。

TempKey 寄存器的数据部分配有一组标志，指示有关数据源及其有效性的信息。Info 命令可用于返回此寄存器对应的某些状态/标志位的值，如下所示：

表 3-1. TempKey 标志

名称	长度	说明
KeyID	4 位	如果 TempKey 由 GenDig 或 GenKey 生成，则这些位指示在其计算中使用哪个密钥。这 4 位表示数据区域的其中一个槽。
SourceFlag	1 位	TempKey 中随机性的来源： 0 = 内部生成的随机数（*Rand*）。 1 = 仅输入（固定）数据，未生成内部随机数（*Input*）。
Generator	4 位	0 = TempKey 不是由 GenDig 生成的。 1 = TempKey 的内容是由 GenDig 使用数据区域中的一个槽生成的（TempKey.KeyID 将是有意义的）。
GenKeyData	1 位	0 = TempKey.KeyID 不是由 GenKey 生成的。 1 = TempKey 的内容是由 GenKey 使用数据区域中的一个槽生成的（TempKey.KeyID 将是有意义的）。
NoMacFlag	1 位	1 = TempKey 的内容是使用 SlotConfig.NoMac 为 1 的槽中的值生成的，因此不能被 MAC 命令使用。如果在 TempKey 的计算中使用了多个槽，则在 SlotConfig.NoMac 针对其中任何槽置 1 时，此位便将置 1。
Valid	1 位	0 = TempKey 中的信息无效。 1 = TempKey 中的信息有效。

在本规范中，这些标志通常称为 TempKey.SourceFlag 和 TempKey.GenDigData 等。当 TempKey.Valid 为 0 时，无论其他标志位的状态如何，任何尝试使用 TempKey 寄存器内容的操作都会导致错误。

在上电、休眠、欠压、看门狗超时或篡改检测期间，TempKey 寄存器及其所有标志都将清零。当器件进入空闲模式时，TempKey 的内容以及标志将保留不变。

通常，每当在命令执行期间出于任何目的使用（读取）TempKey 时，TempKey.Valid 和所有其他标志都将清零。当要使用 TempKey 的命令遇到错误时，可能会也可能不会清零 TempKey，具体视情况而定。如果特定命令或命令模式/配置未使用 TempKey，则其始终不会清零。TempKey 始终不会被 KDF 或 AES 命令清零。

将结果保留在 TempKey 中的命令会将 Valid 标志以及可能适用于所执行操作的任何其他标志置 1。

3.2 报文摘要缓冲区

报文摘要缓冲区是一个 64 字节寄存器，在需要通过 TempKey 寄存器来保留其他信息时，用于将输入报文摘要传送到 Verify 和 Sign 命令。SHA 命令可将摘要直接写入该寄存器，以简化外部主机编程。

如果 Verify 命令的输出需要验证 MAC，该寄存器始终用于传送计算该 MAC 所用的 nonce。nonce 在报文摘要缓冲区中的位置取决于是通过 TempKey 还是报文摘要缓冲区输入签名报文摘要。

Nonce 命令可以将 32 或 64 字节的固定输入数据写入报文摘要缓冲区。

在上电、休眠、欠压、看门狗超时或篡改检测期间，报文摘要缓冲区将清零。通常，在执行完大多数命令（Nonce 和 SHA 命令除外）后，报文摘要缓冲区会清零。它只能在单个命令中使用（读取），而无需重载，因为其始终在使用后清零。

3.3 备用密钥缓冲区

备用密钥缓冲区是一个 32 字节寄存器，在需要通过 TempKey 寄存器来保留其他信息时，供 KDF 命令用于存储密钥。可通过 Nonce 命令对其写入固定输入值，或通过 KDF 命令对其写入机密信息值。

在上电、休眠、欠压、看门狗超时或篡改检测期间，备用密钥缓冲区将清零。

备用密钥缓冲区的用途是从单个根密钥生成两个单独的基于 SRAM 的密钥。为此，一种实现方法是使用 KDF 命令，输入设置为 AltKeyBuf，输出设置为 TempKey(Lo)。之后，再次运行 KDF，输出设置为 TempKey(Hi)，从而将两个不同的密钥存储在一个存储单元，在本例中为 TempKey。TLS 1.3 可能需要类似的流程。

3.4 SHA 上下文缓冲区

SHA 命令使用标准三阶段流程：初始化、更新和完成。在许多情况下，更新阶段会多次运行。内部 SRAM 存储器用于存储这些阶段之间的中间状态（也称为 SHA 上下文）。

SHA 上下文缓冲区不能被任何其他 ATECC608A-TNGTLS 命令读取或写入，因此无论任何其他命令的执行成功与否，都不会被破坏。与器件中的所有 SRAM 存储器一样，在上电、欠压、掉电、看门狗超时或篡改检测期间，它将清零。

4. 一般命令信息

以下各节介绍了有关基本 I/O 事务、命令结构、错误代码、存储器寻址以及 ATECC608A-TNGTLS 中使用的密钥和签名格式的一般信息。

4.1 I/O 事务

ATECC608A-TNGTLS 利用 I^2C 协议与主机单片机通信。在采用以下方式构建的事务中，安全命令将发送到器件，并从器件接收响应：

表 4-1. I/O 事务格式

字节	名称	含义
0	Count	要传入（或传出）器件的组中的字节数，其中包括计数字节，数据包字节和校验和字节。因此计数字节的值应始终为（N + 1），其中 N 等于数据包中的字节数加上 2 个校验和字节。对于具有 1 个计数字节、50 个数据包字节和 2 个校验和字节的组，应将计数字节设置为 53。最大的组（与计数值）为 155 个字节，最小的组为 4 个字节。如果值超出此范围，将导致器件返回 I/O 错误。
1 至（N-2）	Packet	命令、参数和数据或响应。有关一般命令数据包的信息，请参见 4.2 命令数据包；有关每个命令的具体参数，请参见 5. 器件命令。
N-1 和 N	Checksum	计数字节和数据包字节的 CRC-16 验证。CRC 多项式为 0x8005。在开始 CRC 计算之前，将 CRC 寄存器初始化为零。在计数和数据包的最后一位发送后，内部 CRC 寄存器的值必须与块中的校验和字节相匹配。发送的第一个 CRC 字节（N-1）是 CRC 值的最低有效字节，因此此组的最后一个字节是 CRC 的最高有效字节。

ATECC608A-TNGTLS 的设计应使输入组中的计数值与命令参数中指定的大小要求一致。如果计数值与数据包内的命令操作码和/或参数不一致，ATECC608A-TNGTLS 将根据具体命令以不同的方式响应。响应可能包含错误指示，也可能默默忽略一些输入字节。

4.2 命令数据包

表 4-2 对命令数据包进行了分解：

表 4-2. 命令数据包

字节	名称	含义
0	Opcode	命令代码。请参见 5. 器件命令
1	Param1	第 1 个参数，始终存在。
2-3	Param2	第 2 个参数，始终存在。
0-155	Data	可选的剩余输入数据。

在 ATECC608A-TNGTLS 接收到组中的所有字节之后，器件将转换到繁忙状态并尝试执行命令。当器件繁忙时，无法从其中读取状态和结果。在此期间，器件的 I/O 接口会忽略 I^2C SDA 输入信号的所有转换。

4.3 状态/错误代码

器件没有专用的状态寄存器，因此状态、错误和命令结果共用输出 FIFO。器件的所有输出都将以完整组的形式返回到系统，这些组的格式与输入组相同：

- 计数

- 数据包
- 2 字节 CRC

器件接收到输入命令组的第 1 个字节后，系统将无法读取器件中的任何内容，直到系统将所有字节发送给器件。

在唤醒和执行命令后，器件的输出寄存器中会有错误、状态或结果字节，可通过系统获取。当此组的长度是 4 个字节时，返回的代码详见表 4-3。有些命令执行成功时会返回超过 4 个字节。得到的数据包说明在 5. 器件命令列出。

CRC 错误始终在任何其他类型的错误之前返回。它们表明发生了某种 I/O 错误，并且此命令可重新发送给器件。如果还发生了多个其他错误，则这些错误没有特定的优先顺序。

表 4-3. 4 字节组中的状态/错误代码

状态说明	错误/状态	说明
命令执行成功	0x00	命令执行成功。
Checkmac 或 Verify 不匹配	0x01	CheckMac 或 Verify 命令已正确发送到器件，但输入响应与预期值不匹配。
解析错误	0x03	命令已正确接收，但长度、命令操作码或参数非法，而与 ATECC608A-TNGTLS 的状态（易失性存储器和/或 EEPROM 配置）无关。命令位的值必须在重新尝试之前进行更改。
ECC 故障	0x05	ECC 处理期间发生计算错误，导致结果无效。重试命令可能会使执行成功。
自检错误	0x07	发生自检错误，芯片处于故障模式，等待清除故障。
运行状况测试错误	0x08	发生随机数发生器运行状况测试错误，并且在清除该错误之前，芯片无法执行需要随机数的后续命令。
执行错误	0x0F	命令已正确接收，但无法由器件在当前状态下执行。器件状态或命令位的值必须在重新尝试之前进行更改。
在唤醒之后、第 1 条命令之前	0x11	指示 ATECC608A-TNGTLS 已收到适当的 Wake 令牌。
看门狗即将超时	0xEE	在看门狗定时器超时之前没有足够的时间执行给定的命令。系统必须通过进入空闲或休眠模式来复位看门狗定时器。
CRC 或其他通信错误	0xFF	命令未由 ATECC608A-TNGTLS 正确接收，应由系统中的 I/O 驱动器重新发送。未尝试解析或执行命令。

4.4 地址编码

以下各小节提供了有关如何寻址 ATECC608A-TNGTLS 器件的各个存储区域的详细信息。

4.4.1 配置区域寻址

对于配置区域，可一次性访问 4 或 32 个字节。无法访问单个字节。配置区域地址为 2 字节（16 位值）。配置区域寻址仅使用地址字的低 5 位。对于 ATECC608A-TNGTLS 器件，这些地址只能与读命令搭配使用。

表 4-4. 地址格式

字节 1：Addr[15:8]	字节 0：Addr[7:0]		
未使用	未使用	块	偏移量
Addr[15:8]	Addr[7:5]	Addr[4:3]	Addr[2:0]

表 4-5. 配置区域地址

块号（Addr[4:3]）	偏移值（Addr[2:0]）							
	000	001	010	011	100	101	110	111
00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
01	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
10	[64:67]	[68:71]	[72:75]	[76:79]	[80:83]	[84:87]	[88:91]	[92:95]
11	[96:99]	[100:103]	[104:107]	[108:111]	[112:115]	[116:119]	[120:123]	[124:127]

4.4.2 OTP 区域寻址

对于可一次性编程（OTP）区域，可一次性访问 4 或 32 个字节。该区域总共有 64 个字节。无法访问单个字节。OTP 区域地址为 2 字节（16 位值）。仅低四位用于寻址。

对于 ATECC608A-TNGTLS 器件，这些地址只能与读命令搭配使用。

表 4-6. 地址格式

字节 1：Addr[15:8]	字节 0：Addr[7:0]		
未使用	未使用	块	偏移量
Addr[15:8]	Addr[7:4]	Addr[3]	Addr[2:0]

表 4-7. OTP 区域字节地址

块号（Addr[3]）	块偏移值（Addr[2:0]）							
	000	001	010	011	100	101	110	111
0	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
1	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]

4.4.3 数据区域寻址

与配置区域和 OTP 区域相比，对数据区域的读/写访问要复杂得多。数据区域总共有 16 个槽，槽的大小各不相同。每个槽的访问策略分别控制一个槽是否允许读取或写入。

对于 ATECC608A-TNGTLS：

- 数据 Slot 6 和 8-9 可采用明文形式写入。
- 任何数据槽均无法采用密文形式写入。
- 数据 Slot 8 和 10-12 可采用明文形式读取。
- 任何未指定的槽均无法读取或写入。

表 4-8. 基于数据槽大小的地址格式

数据区域	字节 1 Addr[15:8]		字节 0：Addr[7:0]		
	未使用	块	未使用	槽	偏移量
数据槽[7:0]	Addr[15:9]	Addr[8]	Addr[7]	Addr[6:3]	Addr[2:0]
数据槽[8]	Addr[15:12]	Addr[11:8]	Addr[7]	Addr[6:3]	Addr[2:0]
数据槽[15:9]	Addr[15:10]	Addr[9:8]	Addr[7]	Addr[6:3]	Addr[2:0]

数据槽[7:0]
要完全访问这些槽之一，需要进行两次 32 字节访问或 9 次 4 字节访问

表 4-9. 数据区域寻址 Slot 0-7

槽号（Addr[6:3]）	块号（Addr[8]）	块偏移值（Addr[2:0]）							
		000	001	010	011	100	101	110	111
0x0 至 0x7	00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	01	[32:35]	无效	无效	无效	无效	无效	无效	无效

数据槽[8]
要完全访问此槽，需要进行 13 次 32 字节访问或 104 次 4 字节访问，或结合使用这两种方法。

表 4-10. 数据区域寻址 Slot 8

槽号（Addr[6:3]）	块号（Addr[8]）	块偏移值（Addr[2:0]）							
		000	001	010	011	100	101	110	111
0x8	0x0	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	0x1	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
	...	...	...	...	...	...	...	...	...
	0xC	[384:387]	[388:391]	[392:395]	[396:399]	[400:403]	[404:407]	[408:411]	[412:415]

数据槽[15:9]
要完全访问这些槽，需要进行 3 次 32 字节访问或 18 次 4 字节访问，或结合使用这两种方法。

表 4-11. 数据区域寻址 Slot 9-15

槽号（Addr[6:3]）	块号（Addr[8]）	块偏移值（Addr[2:0]）							
		000	001	010	011	100	101	110	111
0x9 至 0xF	00	[0:3]	[4:7]	[8:11]	[12:15]	[16:19]	[20:23]	[24:27]	[28:31]
	01	[32:35]	[36:39]	[40:43]	[44:47]	[48:51]	[52:55]	[56:59]	[60:63]
	10	[64:67]	[68:71]	无效	无效	无效	无效	无效	无效

4.5 密钥、签名和证书的格式

以下各节提供了有关 ECC 密钥、签名和压缩证书的详细格式信息。

4.5.1 ECC 密钥格式

公钥和私钥的格式取决于命令和密钥的长度。通常，最高有效字节（MSB）首先出现在总线上的存储器最低地址处。在本节剩余部分中，左侧的字节为 MSB。Microchip 建议将所有填充字节设置为 0 以保持一致性。

- ECC 私钥对用户来说仅仅为 `PrivWrite` 命令的输入参数。此参数的长度始终为 36 个字节，前 4 个字节（32 位）全部为填充位。
  ECC 公钥作为几个命令的输入或输出参数出现，也可以存储在 EEPROM 中。其组成方式为：总线上或存储器中的 X 值在前，后跟 Y 值。它们的格式有所不同，具体取决于如下所述的情况：
- **公钥是 `GenKey` 命令的输出或 `Verify` 命令的输入：**
  先是 32 个 X 字节，随后是 32 个 Y 字节。（36 个字节）没有填充字节。
- **写命令：**

公钥可使用写命令直接写入 EEPROM，其长度始终为 72 字节，格式如下：4 个填充字节，32 个 X 字节，4 个填充字节，32 个 Y 字节。

- **GenKey 命令：**
  SHA 报文：可通过 GenKey 命令对公钥进行哈希运算并将其置于 TempKey 中。SHA 报文包含与密钥大小无关的各个字节。其后依次为 25 个填充字节，32 个 X 字节，32 个 Y 字节。
- **Verify 命令：**
  SHA 报文：当用于验证存储的公钥时，Verify 命令需要通过存储器中存储的密钥的 SHA-256 摘要创建的输入签名。此类内部 SHA 计算始终通过 72 个字节来执行，这些字节在 EEPROM 中的存储格式为 4 个填充字节、32 个 X 字节、4 个填充字节，随后是 32 个 Y 字节。

当公钥配置为通过 Verify 命令验证时，存储器中第一个字节的高 4 位供器件在内部用来保存验证状态。它们始终由写命令设置为无效状态（0xA），之后可通过 Verify 命令设置为有效状态（0x5）。

下面将介绍 I/O 协议的最低层。在 I/O 协议层之上，完全相同的字节传入和传出器件以实现命令。后续章节将对错误代码进行说明。

4.5.1.1 公钥格式

ATECC608A-TNGTLS 支持两种格式的 P-256 椭圆曲线公钥。以下示例详细说明了这两种格式。

对于以下示例，我们将使用示例公钥，其中 X 和 Y 整数以固定宽度的大尾数无符号整数表示：

```
X：b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
Y：a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

命令公钥格式
任何返回公钥（GenKey）或接受公钥作为参数的命令（Verify 和 ECDH）都会将公钥格式化为 X 和 Y 大尾数无符号整数连在一起的形式，因而总共 64 个字节。

例如：

```
b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

存储的公钥格式
将公钥存储在槽中以便与 Verify 或 SecureBoot 命令搭配使用时，X 和 Y 整数将分别被填充为两个 36 个字节并连在一起，因而总共为 72 个字节。

例如：

```
00000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

注： 仅 Slot 8-15 便足以保存公钥。

存储的经验证公钥格式
经验证和未经验证的公钥格式与存储的公钥格式相同，但 LSB 的高四位除外。如果密钥经验证，则最低有效半字节将为 0x5，如果未经验证则为 0xA。这些值可在验证或未验证模式下通过 Verify 命令进行更改。写入的密钥最初为未验证状态。

经验证公钥示例：

```
50000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

未验证公钥示例：

```
A0000000b2be345ad7899383a9aab4fb968b1c7835cb2cd42c7e97c26f85df8e201f3be8
00000000a82983f0a11d6ff31d66ce9932466f0f2cca21ef96bec9ce235b3d87b0f8fa9e
```

注： 仅 Slot 8-15 便足以保存公钥。

4.5.2 签名格式

无论是由 Sign 命令生成并输出的 ECDSA 签名还是输入到 Verify 或 SecureBoot 命令的 ECDSA 签名，长度始终为 64 字节。签名分为 R 和 S 两个组成部分。这两者的长度均为 32 字节，并且在总线上 R 始终出现在 S 之前。签名的每个部分在总线上都先显示 MSB，这意味着签名的 MSB 位于最低存储单元中。

R/S 签名示例

任何返回签名（Sign）或接受签名作为参数的命令（Verify 和 SecureBoot）都会将签名格式化为 R 和 S 大尾数无符号整数连在一起的形式，因而总共 64 个字节。

例如：

```
R: 7337887F8C39DF79FD8BF88DDFBFB9DB15D7B1AD68196AE3FB0CE5BFA2842DF3
S: 72868A43A42831E950E1DA9F73B29F5C0ED8A96B2889E3CBBE8E61EA6C67F673
```

4.5.3 证书存储

器件内的完整 X.509 证书所需的存储空间较大，可能会很快占满多个 EEPROM 存储器槽。根据实际应用的不同，这些槽也并非一定适合用来存储证书。为了应对这些存储限制，Microchip 定义了一种编码方式，以便能够基于少量信息重建完整的 X.509 证书。

主机系统实际上将负责重建完整的 X.509 证书，但是如何执行将由存储在编码证书中的数据决定。对于给定系统，所有器件共用的数据可轻松存储在主机系统中。其他数据可根据已存储在器件中的数据轻松进行计算或提取。 表 4-12 指示 X.509 证书中存储的数据类型，以及如何对其进行编码以便能够存储到单个 72 字节槽中。

表 4-12. 证书存储

X.509 证书		编码证书		
X.509 元素	**大小（字节）**	**编码证书元素**	**器件证书（位）**	**签名人证书（位）**
序列号	8-20	序列号来源	4	4
颁发日期	13	压缩格式	19	19
有效期	13	有效年数	5	5
签名人 ID(2)	4	用于对证书（器件证书）进行签名的特定签名人的 ID 或签名人本身（签名人证书）的 ID	16	16
AuthorityKeyIdentifier	20	颁发机构公钥的 SHA1 哈希	0	0
SubjectKeyIdentifier	20	主题公钥的 SHA1 哈希	0	0
签名 R	32	存储在器件中	256	256
签名 S	32	存储在器件中	256	256
公钥 X(1)	32	根据私钥计算或存储在器件中(1)	0	256
公钥 Y(1)	32	根据私钥计算或存储在器件中(1)	0	256
n/a	0	证书格式	4	4
n/a	0	模板 ID	4	4
n/a	0	链 ID	4	4
n/a	0	保留/用户自定义	8	8
总计	（206-218 字节）		576 位（72 字节）	1088 位（136 字节）

注：

1. 对于器件证书，可以根据私钥重新生成器件公钥。对于签名人证书，公钥通常存储在单独的槽中。
2. 对于器件证书，将存储用于对证书进行签名的签名人的 ID。对于签名人证书，将存储签名人的实际 ID，以便设备能够识别。

Slot 8 总共包含 416 个字节。根据证书中存储的序列号的大小，可能会也可能不会存储两个完整的证书。通常，在已创建信任链的器件中，器件证书、签名人证书和签名人公钥必须存储在器件中。

更多信息，请参见 Compressed Certificate Definition 应用笔记。

5. 器件命令

以下部分详细介绍了 ATECC608A-TNGTLS 中允许的基于命令模式细分的所有命令。这些命令分为三类：

1. **常规器件命令**
   此类命令分为两类：
   - 常规器件访问命令，用于将数据发送到器件或检索数据，但通常不执行任何加密功能。
   - 常规加密命令，可由器件或系统使用，但通常不会用于特定数据槽。
2. **非对称加密命令**
   这些命令执行非对称加密操作，例如密钥生成、报文签名和报文验证（使用 ECC 公钥或私钥）。这些命令仅限用于 ECC 数据区域槽。
3. **对称加密命令**
   这些命令执行对称加密功能，例如生成摘要或 MAC、密钥派生或 AES 加密和解密。

所有命令的输入参数
除非另有说明，否则输入参数表中的多字节输入参数以大尾数值的形式（MSB 在前）显示。请注意，ATECC608A-TNGTLS 器件实际上希望数据以小尾数形式（LSB 在前）发送。

5.1 常规器件命令

下表总结了常规器件命令：

表 5-1. 常规器件命令

命令名称	操作码	说明
Counter	0x24	递增并读取单调计数器。
Info	0x30	用于从器件中读取版本和状态信息。
Lock	0x17	用于锁定器件中的各个可锁定槽。
Nonce	0x16	用于为器件生成或传递仅使用一次的数字。
Random	0x1B	用于生成供系统使用的 32 字节随机数。
Read	0x02	用于读取器件的各个区域。
SelfTest	0x77	测试各种内部加密计算元素。
SHA	0x47	计算系统通用的 SHA-256 或 HMAC 摘要。
UpdateExtra	0x20	配置区域锁定后，更新配置区域内的字节 84 或 85。
Write	0x12	用于向器件写入 4 个或 32 个字节，可以使用或不使用身份验证和加密。

5.1.1 Counter 命令

Counter 命令在配置区域内读取器件上 2 个单调计数器之一的二进制计数值。此计数器可达到的最大值为 2,097,151。尝试计数超出此值将导致错误代码。此计数器的设计可确保即使计数操作期间电源中断，也始终不会丢失计数。在某些电力损失的情况下，计数器可能会以超过 1 的值递增。

对于 ATECC608A-TNGTLS，计数器未关联到任何密钥，但仍可供系统使用。每个计数均设置为其默认值，并且可以计数到最大值。

表 5-2. 输入参数——计数

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	说明
0x24	0x00	0x00 00	读取 Counter[0]的值
		0x00 01	读取 Counter[1]的值
	0x01	0x00 00	使 Counter[0]的值递增
		0x00 01	使 Counter[1]的值递增

表 5-3. 输出响应——计数

名称	大小	说明
Count	4 字节	如果命令成功，则为二进制计数器值。
Response	1 字节	如果命令失败，则为错误代码。

5.1.2 Info 命令

Info 命令用于读取器件的状态。该信息对于确定错误或使用各种命令很有用。

5.1.2.1 信息——版本

Info 命令的版本模式可读回 ATECC608A-TNGTLS 的芯片版本。该信息被硬编码到器件中。该信息可能与配置区域中显示的从 Revision 字节读回的信息相同，也可能有所不同。

表 5-4. 输入参数——信息版本

操作码（1 字节）	模式（1 字节）	参数（2 字节）	数据（0 字节）	说明
0x30	0x00	0x00 00	—	返回芯片版本

表 5-5. 输出响应——信息版本

名称	响应	说明
Response	00 00 60 vv	版本信息。0x60 表示 ATECC608A-TNGTLS。vv 表示最新的芯片版本。

5.1.2.2 Info——KeyValid

KeyValid 模式用于确定存储的 ECC 私钥或公钥是否为有效的 ECC 密钥。如果无法输出 PublicKey，则该命令返回的信息将无用。如果 KeyType 不为 ECC，则该命令的输出也没有用。

对于 ATECC608A-TNGTLS 器件，Slot 1-4 中存储的密钥是 ECC 密钥，可以通过 Info 命令的 KeyValid 模式进行检查。

表 5-6. 输入参数——信息 KeyValid

操作码（1 字节）	模式（1 字节）	参数（2 字节）	数据（0 字节）	说明
0x30	0x01	0x00 0[槽号]	—	返回槽是否包含有效 ECC 私钥或公钥的相关信息。

表 5-7. 输出响应——信息 KeyValid

名称	大小 （4 字节）	说明
Response	0x00 00 00 00	ECC 密钥无效
	0x00 00 00 01	ECC 密钥有效

5.1.2.3 信息——器件状态

在该模式下，使用 Info 命令可返回器件的当前状态。状态位可用于确定器件的当前状态，并且可用于确定给定命令失败的原因或确定命令是否可以执行。

表 5-8. 输入参数——信息器件状态

操作码 （1 字节）	模式 （1 字节）	参数 （2 字节）	数据 （0 字节）	说明
0x30	0x02	0x00 00	—	返回器件状态

表 5-9. 输出响应——信息器件状态

名称	响应	说明
Response	0x00 00 Byte[1] Byte[0]	

表 5-10. 状态标志

字节编号	位号	名称	说明
0	7	TempKey.NoMacFlag	0：NoMacFlag 无效 1：NoMacFlag 有效
	6	TempKey.GenKeyData	0：GenKeyData 无效 1：GenKeyData 有效
	5	TempKey.GenDigData	0：GenDigData 无效 1：GenDigData 有效
	4	TempKey.SourceFlag	0：TempKey 为固定源 1：TempKey 来自随机数源
	3:0	TempKey.KeyID	TempKey 密钥槽 ID
1	7	TempKey.Valid	0：TempKey 无效 1：TempKey 有效
	6:3	AuthComplete.KeyID	授权密钥槽 ID
	2	AuthComplete.Valid	0：授权无效 1：授权有效
	1:0	未使用	2’b00

5.1.3 Lock 命令

对于 ATECC608A-TNGTLS，配置区域已被锁定，数据区域的访问策略已被强制执行。但是，仍然有几个数据槽可以通过使用其他命令来更新。如有需要，可以使用 Lock 命令的槽锁定模式将其中一些槽永久锁定，以防将来更新。

5.1.3.1 SlotLock

Lock 命令的 SlotLock 模式用于单独锁定各个槽。可锁定位置 1 且之前未进行槽锁定的任何槽都可以锁定，以防止任何进一步的更新。此过程不可逆。槽一旦锁定，将被永久锁定。在该工作模式下，将忽略摘要 CRC 和数据字段。

对于 ATECC608A-TNGTLS，以下各槽可能被锁定：Slot 2-4、6 和 Slot 8。

表 5-11. 输入参数——SlotLock

操作码（1 字节）	模式（1 字节）	摘要 CRC（2 字节）	数据（0 字节）	说明
0x17	8’b00[bb_bb]10(1)	0x00 00	—	单独槽锁定

注： 1——[bb_bb]为代表被锁定的槽的四位值。

表 5-12. 输出响应——SlotLock

名称	大小	说明
Response	1 字节	如果命令成功将槽锁定，则为 0x00。如果命令失败，则会输出错误代码。

5.1.4 Nonce 命令

Nonce 命令通过将随机数（可在内部生成，也可在外部生成）与来自系统的输入值相组合来生成 nonce（仅使用一次的数字），以供后续命令使用。得到的 nonce 在内部存储在三个可能的缓冲区中：TempKey 缓冲区、报文摘要缓冲区和备用密钥缓冲区。必要时，可将值传递给器件，而不是生成 nonce。

5.1.4.1 Nonce——随机

Nonce 命令在随机模式下运行时，会根据下表中显示的输入值生成一个新的 nonce。如果 Param2 为 0x00 00，则根据内部 RNG 生成一个新的随机数。如果 Param2 为 0x80 00，则使用 TempKey 中存储的值来生成新的 nonce，并且不运行随机数发生器。在这种情况下，TempKey 必须有效，然后才能运行 Nonce 命令。完成命令后，TempKey.SourceFlag 将设置为 Rand。

建议从随机数源生成发送到器件的 20 字节数据，以防止重放攻击。

表 5-13. 输入参数——随机 Nonce

操作码	模式	Param2	数据	说明
0x16	0x00 或 0x01	0x00 00	20 字节	• 写入 TempKey 的 32 字节 • 输出来自 RNG • SHA256 摘要包含随机数
	0x00 或 0x01	0x80 00	20 字节	• 写入 TempKey 的 32 字节 • 输出为 SHA256 哈希值 • SHA256 摘要包含 TempKey

表 5-14. 输出响应——随机 Nonce

名称	输入 Param2	大小	说明
Response	0x00 00	32 字节	随机数
	0x80 00	32 字节	新 TempKey 值

注：

1. 成功完成命令后，TempKey.SourceFlag 设置为 0，以指示 nonce 值为随机数源。
2. 成功时 TempKey.Valid 置 1。

表 5-15. Nonce SHA256 哈希计算

字节数	输入数据 Param2 = 0x00 00	输入数据 Param2 = 0x80 00
32	来自随机数发生器的 RandOut	来自上一条命令的 TempKey 值
20	来自输入流的 NumIn	来自输入流的 NumIn
1	操作码（始终为 0x16）	操作码（始终为 0x16）
1	模式（0x00 或 0x01）	模式（0x00 或 0x01）
1	Param2 的 LSB（始终为 0x00）	Param2 的 LSB（始终为 0x00）

5.1.4.2 Nonce——固定

固定 nonce 被传递到器件并存储在内部缓冲器之一中。nonce 的大小可以是 32 字节或 64 字节。运行此命令后，TempKey.SourceFlag 始终设置为输入。这种 Nonce 命令模式不会进行 SHA256 计算或生成随机数。

表 5-16. 输入参数——固定 Nonce

操作码	模式	Param2	数据	说明
0x16	0x03 0x43 0x83	0x00	32 字节	• 写入 TempKey 的 32 字节 • 写入报文摘要缓冲区的 32 字节 • 写入备用密钥缓冲区的 32 字节
	0x23 0x63	0x00	64 字节	• 写入 TempKey 的 64 字节 • 写入报文摘要缓冲区的 64 字节

表 5-17. 输出响应——固定 Nonce

名称	大小	说明
Response	1 字节	如果命令成功完成，则为 0x00。否则，将收到错误代码。

注：

1. 成功完成命令后，TempKey.SourceFlag 设置为 1，以指示 nonce 为固定的输入值。
2. TempKey.Valid 设置为 1，以指示值可用于其他用途。

5.1.5 Random 命令

Random 命令生成一个随机数，以供系统使用。随机数通过内部 NIST 800-90 A/B/C 随机数发生器生成。命令的输出始终是总线上的 32 字节数字。该数字不能存储在任何数据槽或 SRAM 存储单元中。

表 5-18. 输入参数——随机

操作码 （1 字节）	模式 （1 字节）	Param2 （2 字节）	数据 （0 字节）	说明
0x1B	0x00	0x00 00	—	Random 命令

表 5-19. 输出响应——随机

名称	大小	说明
RandOut	32 字节	RNG 的输出

5.1.6 Read 命令

读命令可用于访问 ATECC608A-TNGTLS 器件的任何 EEPROM 区域。根据为每个槽设置的访问策略来限制数据区域访问。如果设置了特殊访问策略，则只能在数据区域槽进行加密读取。

5.1.6.1 明文读取

始终可从 ATECC608A-TNGTLS 器件的配置区域和 OTP 区域读取明文。 可根据已设置的访问策略，以明文形式读取数据区域中的特定槽。这些区域均支持 4 字节或 32 字节读取。

对于 ATECC608A-TNGTLS 器件，Slot 8 和 10-12 包含可采用明文形式读取的数据。

表 5-20. 输入参数——明文读取

操作码（1 字节）	模式（1 字节）	地址（2 字节）	说明
0x02	0x00	请参见 4.4 地址编码。	4 字节配置区域读取
	0x80	请参见 4.4 地址编码。	32 字节配置区域读取
	0x01	请参见 4.4 地址编码。	4 字节 OTP 区域读取
	0x81	请参见 4.4 地址编码。	32 字节 OTP 区域读取
	0x02	请参见 4.4 地址编码。	4 字节数据区域读取
	0x82	请参见 4.4 地址编码。	32 字节数据区域读取

表 5-21. 输出响应——明文读取

名称	大小	说明
Data Contents	4 字节	已发送 4 个字节[0:3]
	32 字节	已发送 32 个字节[0:31]

5.1.7 SelfTest 命令

SelfTest 命令对 ATECC608A-TNGTLS 芯片中的一个或多个加密引擎进行测试。将根据输入模式参数测试部分或全部算法。

对于 ATECC608A-TNGTLS 器件，SelfTest 命令已在上电或唤醒事件后被自动禁止，无法运行。但是，如有需要，可以通过系统执行命令。无需运行此测试。

如果任何自检失败，则无论是在上电和唤醒时自动调用还是通过此命令调用，芯片都会进入故障状态，在该状态下芯片操作受限。唤醒或电源重启后，始终会清除存储的故障状态。请注意，自检失败（错误代码：0x07）不同于运行状况测试失败（错误代码：0x08）。

在故障状态下，允许以下操作：

- 读取配置区域。
- 执行此自检命令。如果后续尝试重新进行的特定测试通过，则将清零故障寄存器中的该位。如果所有位均清零，则 ATECC608A-TNGTLS 会恢复正常命令操作。
- 可以通过模式参数 0 调用此自检命令来读取故障寄存器的当前状态。
- 执行任何其他命令或读取任何其他区域都将返回错误代码 0x07。使用 SelfTest(0)确定失败的原因

表 5-22. 输入参数——SelfTest

操作码（1 字节）	模式（1 字节）[1]							Param2（2 字节）
	b[7:6]	b[5]	b[4]	b[3]	b[2]	b[1]	b[0]	
	2’b00	SHA	AES	ECDH	ECDSA（签名和验证）	0	RNG 和 DRBG	0x00 00

注：

1. 可以一次性进行任意测试组合。将相应模式位置 1 即表示将进行测试。如果相应位为 0，则不会进行该测试。

表 5-23. 输出响应——SelfTest

名称	大小	说明
Success	1 字节	0x00——所有测试均通过 失败映射——1 表示测试失败。失败位与模式字节中的位相对应。

5.1.8 SHA 命令

SHA 命令用于计算主机系统通用的 SHA-256 或 HMAC/SHA 摘要。SHA 计算在内部 ATECC608A-TNGTLS 存储器中未被任何其他命令读取和写入的特殊部分（上下文缓冲区）中执行。任意命令都可以穿插在 SHA 命令各个阶段之间，而不会产生问题。上电和唤醒时会使该 SHA 上下文失效。在大多数情况下，如果执行 SHA 命令时发生错误，上下文将保持不变。

5.1.8.1 SHA——SHA256 摘要

SHA 命令利用 SHA256 哈希算法来对报文进行哈希计算。该命令的主输入为实际报文。报文以 1 到 64 字节块的形式提交给器件。无需上下文切换时，使用以下步骤：

1. 在启动模式下发出 SHA 命令。不包含任何报文。
2. 在更新模式下，以 1 到 64 字节报文的形式发出 SHA 命令。
3. 重复步骤 2，直到提交的报文字节数达到所需字节总数。
4. 在结束模式下发出 SHA 命令，以完成 SHA256 计算。

表 5-24. 输入参数——SHA 标准模式

操作码（1 字节）	模式（1 字节）	Param2（2 字节）	数据（因模式而异）	说明
0x47	0x00	0x00 00	0 字节	启动模式
	0x01	0x00 [字节计数]	1 到 64 字节	更新模式
	0x02	0x00 [字节计数]	0 到 64 字节	完成模式： 摘要置于输出缓冲区和 TempKey 中
	0x42	0x00 [字节计数]		完成模式：摘要置于输出缓冲区和报文摘要缓冲区中
	0xC2	0x00 [字节计数]		完成模式：摘要仅置于输出缓冲区中

表 5-25. 输出响应——SHA256 标准

名称	模式	大小	说明
Response	0x00 和 0x01	1 字节	成功时为 0x00。否则，将收到错误代码
	0x02、0x42 和 0xC2	1 字节 32 字节	错误代码 SHA256 摘要

5.1.8.2 SHA——HMAC 摘要

SHA 命令可用于计算 HMAC 摘要，而不是 SHA256 摘要。二者的计算过程本质上是相同的，只是前者需要使用内部密钥和 HMAC_START（而非 START）。

无需上下文切换时，使用以下步骤：

1. 在 HMAC_Start 模式下发出 SHA 命令，其中指示了密钥位置。不包含任何报文。
2. 在更新模式下，以 1 到 64 字节报文的形式发出 SHA 命令。
3. 重复步骤 2，直到提交的报文字节数达到所需字节总数。
4. 在结束模式下发出 SHA 命令，以完成 HMAC 摘要计算。

表 5-26. 输入参数——SHA HMAC 模式

操作码（1 字节）	模式（1 字节）	Param2（2 字节）	数据（因模式而异）	说明
0x47	0x04	0x00 0[槽号]	0 字节	HMAC 启动模式，并包含来自数据槽的密钥。
	0x04	0xFF FF	0 字节	HMAC 启动模式，并包含来自 TempKey 的密钥
	0x01	0x00 [字节计数]	1 到 64 字节	更新模式
	0x02	0x00 [字节计数]	0 到 64 字节	完成模式： 摘要置于输出缓冲区和 TempKey 中
	0x42	0x00 [字节计数]		完成模式：摘要置于输出缓冲区和报文摘要缓冲区中
	0xC2	0x00 [字节计数]		完成模式：摘要仅置于输出缓冲区中

表 5-27. 输出响应——SHA256 标准

名称	模式	大小	说明
Response	0x04 和 0x01	1 字节	成功时为 0x00；否则，将收到错误代码
	0x02、0x42 和 0xC2	1 字节 32 字节	错误代码 成功时为 HMAC 摘要

5.1.8.3 SHA——上下文切换

上下文切换功能可中断摘要的生成过程，转为执行其他功能或生成其他摘要。上下文切换只能用于 SHA256 摘要模式，因此只能在发出 SHA `Start` 命令之后和 SHA `Finalize` 命令之前进行。在摘要生成过程中，上下文切换可能会发生多次。

上下文切换涉及两个阶段：

1. Read_Context——从 ATECC608A-TNGTLS 读取可变长度上下文，同时使上下文在芯片内有效。输出数据参数的总长度始终为 40 到 99 个字节，具体可以根据输出数据包中的长度字段来确定，也可以通过将输出中第一个字节的低 6 位与 40 相加来计算。
2. Write_Context——将主机中的 SHA256 上下文写入 ATECC608A-TNGTLS，以便完成后续更新操作。该上下文必须事先使用 Read_Context 模式从芯片读取。ATECC608A-TNGTLS 通过数据参数的前 4 个字节确定上下文大小。

读取上下文后，器件可根据需要执行任何其他操作。完成其他操作后，可以将上下文写回到 ATECC608A-TNGTLS 并且 SHA256 摘要生成过程可能会继续进行，直到完成为止。

表 5-28. 输入参数——SHA 上下文切换

操作码（1 字节）	模式（1 字节）	Param2（2 字节）	数据（因模式而异）	说明
0x47	0x06	0x00 00	0 字节	读取当前上下文
	0x07	0x00 [字节计数]	40 到 99 字节	从上一个会话还原当前上下文

表 5-29. 输出响应——SHA 上下文切换

名称	模式	大小	说明
Response	0x06	1 字节 40-99 字节	错误代码 上下文值
	0x07	1 字节	成功时为 0x00；否则，将收到错误代码

5.1.9 UpdateExtra 命令

UpdateExtra 命令用于更新 UpdateExtra 和 UpdateExtraAdd 字节（在配置区域中分别为 Byte 84 和 85）。这些字节只能通过此命令进行更新。这些字节是可一次性更新字节，只有在当前值为 0x00 时才能更新。如果值不为 0x00，则尝试更新此字节将导致错误。

对于 ATECC608A-TNGTLS 器件，UpdateExtraAdd 字节（字节 85）已配置为备用 I^2C 地址。

表 5-30. 输入参数——UpdateExtra

操作码（1 字节）	模式（1 字节）	Param2（2 字节）	数据（0 字节）	说明
0x20	0x00	0x00 [值]	—	使用 Param2 的 LSB 中的值写入 UpdateExtra 字节（Byte 84）。
—	0x01	0x00 [值]	—	使用 Param2 的 LSB 中的值写入 UpdateExtraAdd 字节（Byte 85）。

表 5-31. 输出响应——UpdateExtra

名称	大小	说明
Success	1 字节	0x00——字节成功写入。如果字节未成功写入，则会收到错误代码。

5.1.10 Write 命令

对于 ATECC608A-TNGTLS，配置区域和 OTP 区域已被锁定，无法对这些区域进行更新。数据区域上的写入功能有限，具体取决于每个槽的访问策略。该命令的子模式中描述了可以写入的槽。

5.1.10.1 数据区域——明文写入

标准明文写入

只有在数据区域槽未处于 SlotLocked 模式下时，才允许对其进行明文写入。请注意，任何给定的槽都可允许基于槽的大小将多个数据块写入其中。如果槽不是机密的，则允许对槽内的任何块进行 4 或 32 字节的写操作。对于机密信息槽，仅允许 32 字节写操作。任意槽的最后一块都不是 32 个字节。但这不影响进行 32 字节写操作，只需为其他字节填充零即可。对于 ATECC608A-TNGTLS 器件，Slot 6 和 8-9 可采用明文形式写入。

表 5-32. 输入参数——明文写入

操作码（1 字节）	模式（1 字节）	地址（2 字节）	数据（4 或 32 字节）	说明
0x12	0x02	请参见 4.4 地址编码	4 字节	4 字节写操作
	0x82	请参见 4.4 地址编码	32 字节	32 字节写操作

表 5-33. 输出响应——明文写入

名称	大小	说明
Response	1 字节	如果成功，它将返回值 0x00。如果不成功，则将返回错误代码。

5.2 非对称加密命令

非对称加密命令集由专门用于生成或使用 ECC 密钥的命令组成。密钥通常存储在数据区域槽中，但对于某些命令，也可以存储在 SRAM 阵列中。

表 5-34. 非对称加密命令

命令名称	操作码	说明
ECDH	0x43	使用存储的私钥和输入公钥生成 ECDH 预主机密信息。
GenKey	0x40	生成 ECC 私钥，还可通过存储的私钥生成 ECC 公钥。
Sign	0x41	使用 ECC 私钥和 ECDSA 签名计算对内部或外部报文摘要进行签名。
Verify	0x45	使用 ECC 公钥和 ECDSA 验证计算对内部或外部报文摘要进行验证。

5.2.1 ECDH 命令

ECDH 命令用于生成两个器件之间共享的机密信息。通过从另一器件传递 ECC 公钥，并与存储在槽中的 ECC 私钥或存储在 TempKey 中的临时密钥组合，然后在另一器件上进行相反操作，这两个器件将生成相同的共享主机密信息。然后，可以在两侧将其与其他通用数据进一步组合以生成两个器件之间共享的会话密钥。KDF 命令通常与 TLS 会话搭配使用，以进一步使共享机密信息多样化。

5.2.1.1 ECDH——存储的密钥

ECDH 命令可使用内部数据槽作为其 ECC 私钥源。该槽必须配置其访问策略，以使该槽为 ECC 私钥，并允许 ECDH 命令。访问策略还可指定是否对输出进行存储或加密，或者允许命令本身确定是否对输出进行存储或加密。需要加密时，使用 IO 保护密钥。仅当数据输出到输出缓冲区时才可进行加密。

对于 ATECC608A-TNGTLS，ECDH 命令可以使用存储在 Slot 0 和 2-4 中的 ECC 私钥来运行。

表 5-35. 输入参数——ECDH 存储密钥

操作码（1 字节）	模式（1 字节）	KeyId（2 字节）	数据		说明
			数据 1（32 字节）	数据 2（32 字节）	
0x43	0x0C	0x00 0[槽号]	公钥的 X 分量	公钥的 Y 分量	• 结果发送至输出缓冲区 • 输出采用明文形式(1)
	0x0E	0x00 0[槽号]	公钥的 X 分量	公钥的 Y 分量	• 结果发送至输出缓冲区 • 输出被加密
	0x08	0x00 0[槽号]	公钥的 X 分量	公钥的 Y 分量	• 结果存储在 TempKey 中 • 输出可用于其他操作，但不能直接访问。

注：

1. 当 ChipOptions.ECDHPROT 值为 1 时，在该模式下 ECDH 命令的输出将被加密。对于 ATECC608A-TNGTLS，ECDHPROT 字段设置为 0 并且加密将取决于 ECDH 命令的模式。

表 5-36. 输出响应——ECDH 存储密钥

名称	模式	大小	说明
Response	0x0C 或 0x0E	1 字节	如果命令失败，则为错误代码
Response	0x0C	32 字节	明文形式的共享主机密信息
Response OutNonce	0x0E	32 字节 32 字节	密文形式的共享主机密信息 用于加密的 nonce
Response	0x08	1 字节	成功时为 0x00；否则，将返回错误代码

5.2.1.2 ECDH——TempKey 源

ECDH 命令可使用 TempKey 中的值作为 ECDH 命令的起始值。TempKey 中的 ECC 私钥值必须通过 GenKey 命令生成。TempKey 由 ECDH 命令使用后，TempKey.Valid 标志将复位。如果输出返回到 TempKey 存储单元，则该标志将再次置 1。

表 5-37. 输入参数——ECDH TempKey

操作码（1 字节）	模式（1 字节）	KeyId（2 字节）	数据		说明
			数据 1（32 字节）	数据 2（32 字节）	
0x43	0x0D	0x00 00	公钥的 X 分量	公钥的 Y 分量	• 结果发送至输出缓冲区 • 输出采用明文形式(1)
	0x0F	0x00 00	公钥的 X 分量	公钥的 Y 分量	• 结果发送至输出缓冲区 • 输出被加密
	0x09	0x00 00	公钥的 X 分量	公钥的 Y 分量	• 结果存储在 TempKey 中 • 输出可用于其他操作，但不能直接访问。
	0x05	0x00 0[槽号]	公钥的 X 分量	公钥的 Y 分量	• 结果存储在指定的槽中

注：

1. 当 ChipOptions.ECDHPROT 值为 1 时，在该模式下 ECDH 命令的输出将被加密。对于 ATECC608A-TNGTLS，ECDHPROT 字段设置为 0 并且加密将取决于 ECDH 命令的模式。

表 5-38. 输出响应——ECDH TempKey

名称	模式	大小	说明
Response	0x0D 或 0x0F	1 字节	如果命令失败，则为错误代码
Response	0x0D	32 字节	明文形式的共享主机密信息
Response OutNonce	0x0F	32 字节 32 字节	密文形式的共享主机密信息 用于加密的 nonce
Response	0x05 或 0x09	1 字节	成功时为 0x00；否则，将返回错误代码

5.2.2 GenKey 命令

GenKey 命令用于生成 ECC 私钥和通过私钥生成 ECC 公钥，或者生成公钥摘要。该命令仅适用于专用于 ECC 私钥或公钥的槽。在非 ECC 槽上运行该命令将导致错误。

5.2.2.1 私钥——存储在槽中

GenKey 命令可用于生成 ECC P256 私钥并将其存储在已指定为保存 ECC 私钥的数据槽中。当该命令运行时，还将生成相应的 ECC 公钥。如果槽已锁定，该命令将返回错误。在极少数情况下，会生成无效的 ECC 私钥，这也会导致错误。

对于 ATECC608A-TNGTLS，GenKey 命令只能用于生成 Slot 2、3 和 4 中的私钥。更新这些密钥之前不需要授权。

表 5-39. 输入参数——私钥存储在槽中

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	OtherData（0 字节）	说明
0x40	0x04	0x00 0[槽号]	—	• 私钥存储在[槽号]中 • 生成公钥并在总线上输出
0x40	0x0C	0x00 0[槽号]	—	• 私钥存储在[槽号]中 • 生成公钥并在总线上输出 • 生成公钥摘要并将其存储在 TempKey 中

表 5-40. 输出响应——私钥存储在槽中

名称	大小	响应
Response	1 字节	• 如果命令失败，则为 ECC 错误代码
	64 字节	• 公钥 X 和 Y 协调命令的成功

5.2.2.2 私钥——存储在 TempKey 中

GenKey 命令可用于生成临时 ECC 私钥，并将其置于对写入存储单元没有限制的 SRAM 中。该密钥无法读取，但可由 ECDH 命令使用。

表 5-41. 命令参数

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	OtherData（3 字节）	注释
0x40	0x04	0xFF FF	0x00 00 00	• 私钥存储在 TempKey 中 • 生成公钥并在总线上输出

表 5-42. 输出响应——GenKey 存储在 TempKey 中

名称	大小	响应
Response	1 字节	表示因命令错误而生成的 ECC 错误代码
	64 字节	公钥 X 和 Y 协调命令的成功

5.2.2.3 公钥生成

如果已对槽进行了配置，则可以通过存储的 ECC 私钥重新生成 ECC 公钥，并在总线上输出。此外，如有需要，还可以同时生成公钥摘要并将其存储在 TempKey 中。

对于 ATECC608A-TNGTLS，可以通过 Slot 0-4 中存储的私钥来生成公钥。此外，还可选择生成摘要。

表 5-43. 输入参数——公钥生成

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	OtherData（0 字节）	说明
0x40	0x00	0x00 0[槽号]	—	生成公钥并在总线上输出
0x40	0x08	0x00 0[槽号]	—	• 生成公钥并在总线上输出 • 生成公钥摘要并将其存储在 TempKey 中

表 5-44. 输出响应——公钥生成

名称	大小	响应
Response	1 字节	如果命令失败，则为 ECC 错误代码
	64 字节	公钥 X 和 Y 协调命令的成功

公钥摘要生成
通过对 128 个字节进行 SHA256 哈希运算来生成公钥摘要。

32 字节	TempKey
1 字节	操作码
1 字节	Param1
2 字节	Param2
1 字节	SN[8]
2 字节	SN[0:1]
25 字节	零
64 字节	公钥的 X 和 Y 坐标

5.2.2.4 公钥摘要生成

可以通过存储的 ECC 公钥生成公钥的摘要，并将其存储在 TempKey 中。槽号必须指向存储的公钥，因此该命令仅限于 Slot 8 及更高编号的槽。在运行该命令之前，TempKey 必须为有效值。公钥不会在总线上输出。但是，可以使用读命令读取公钥。请注意，在该模式下，OtherData 中的 3 个字节将用于生成摘要，并且模式和 KeyID 字节将被忽略。

对于 ATECC608A-TNGTLS，可以通过 Slot 11 创建摘要。

表 5-45. 输入参数——公钥摘要生成

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	OtherData（3 字节）	注释
0x40	0x10	0x00 0[槽号]	0x[任意值]	创建公钥摘要并将其存储在 TempKey 中

表 5-46. 输出响应——公钥摘要生成

名称	大小	响应
Response	1 字节	• 如果命令成功完成，则为 0x00 • 如果命令失败，则为 ECC 错误代码

公钥摘要创建
通过对 128 个字节进行 SHA256 哈希运算来生成公钥摘要。

32 字节	TempKey
1 字节	操作码
1 字节	OtherData[0]
2 字节	OtherData[1:2]
1 字节	SN[8]
2 字节	SN[0:1]
25 字节	零
64 字节	公钥的 X 和 Y 坐标

5.2.3 Sign 命令

Sign 命令使用 ECDSA 算法生成签名。由 KeyID 指定的槽中的 ECC 私钥用于生成签名。器件提供多种模式，具体取决于签名的内容。

5.2.3.1 签名——内部报文

内部报文模式下的 Sign 命令用于对内部生成的报文进行签名。该命令计算内部报文摘要，然后使用 ECDSA 签名算法和 KeyID 中指定的 ECC 私钥对摘要进行签名。内部生成的报文必须始终位于 TempKey 中。TempKey 中的值必须使用 GenDig 或 GenKey 命令生成。如果 TempKey 无效，则会出现错误。典型用途包括：

- 对内部生成的随机密钥进行签名。该密钥通常由 GenKey 命令生成。
- GenKey 或 GenDig 命令的输出，前提是输出位于 TempKey 中。

对于 ATECC608A-TNGTLS 器件，只有 Slot 1 能够对内部生成的报文进行签名。

表 5-47. 输入参数——对内部报文进行签名

操作码（1 字节）	模式（1 字节）	KeyId（2 字节）	说明
0x41	0x00 或 0x20	0x00 0[槽号]	序列号不包含在报文摘要计算中
	0x40 或 0x60	0x00 0[槽号]	序列号包含在报文摘要计算中

表 5-48. 输出响应——对内部报文进行签名

名称	大小	说明
Response	1 字节	如果命令失败，则为错误代码
	64 字节	由 R 和 S 值组成的签名

内部报文生成
内部报文基于 55 字节字段生成，如下所示。

字节计数	不包括序列号	包括序列号
32 字节	TempKey(1)	TempKey(1)
1 字节	操作码	操作码
1 字节	模式	模式
2 字节	KeyID	KeyID
2 字节	（TempKeyFlags.KeyID 的）SlotConfig	（TempKeyFlags.KeyID 的）SlotConfig
2 字节	（TempKeyFlags.KeyID 的）KeyConfig	（TempKeyFlags.KeyID 的）KeyConfig
1 字节	TempKeyFlags(2)	TempKeyFlags(2)
2 字节	零	零
1 字节	SN[8] = 0x01	SN[8] = 0x01
4 字节	零	SN[4:7]
2 字节	SN[0:1] = 0x01 0x23	SN[0:1] = 0x01 0x23
2 字节	零	SN[2:3]
1 字节	SlotLocked:TempKeyFlags.KeyID	SlotLocked:TempKeyFlags.KeyID
1 字节	0x00	0x00
1 字节	0x00	0x00

注：

1. 在执行该计算之前，必须先通过 GenKey 或 GenDig 命令生成 TempKey。
2. TempKeyFlags 包括：（b[7]：NoMacFlag，b[6]：GenKeyData，b[5]：GenDigData，b[4]：SourceFlag，b[3:0] TempKeyFlags）

5.2.3.2 签名——外部报文

Sign 命令可用于通过 ECC 私钥对外部报文的摘要进行签名。报文必须由主机系统编译，并且报文摘要必须由主机系统生成。报文可通过 Nonce 命令（在固定模式下运行）装入 TempKey 或报文摘要缓冲区，并且长度始终为 32 个字节。报文始终位于这些存储单元的低 32 个字节中。

对于 ATECC608A-TNGTLS 器件，Slot 0 和 2-4 已使能，以对外部报文进行签名。

表 5-49. 输入参数——外部签名

操作码（1 字节）	模式（1 字节）	KeyId（2 字节）	说明
0x41	0x80 或 0xC0	0x00 0[槽号]	外部报文摘要存储在 TempKey 中
	0xA0 或 0xD0	0x00 0[槽号]	外部报文摘要存储在报文摘要缓冲区中

表 5-50. 输出响应——外部签名

名称	大小	说明
响应	1 字节	如果命令失败，则为错误代码
	64 字节	由 R 和 S 值组成的签名

5.2.4 Verify 命令

Verify 命令采用 ECDSA [R,S]签名，并验证它是否通过给定的输入报文摘要和公钥正确生成。在所有情况下，签名均为命令的输入。

可选 MAC 可通过 Verify 命令返回，以防止所有中间人攻击。如果验证计算表明已通过输入摘要正确生成了签名，则将基于存储在 TempKey 中的输入 nonce 和同时存储在 ATECC608A-TNGTLS 和主机 MCU 中的 IO 保护机密信息值来计算 MAC。MAC 输出只能在外部和存储模式下生成。必须使能 IO 保护功能才能计算 MAC。

5.2.4.1 验证——外部公钥模式

Verify 命令可用于验证在 ATECC608A-TNGTLS 外部生成的报文（其公钥传递给该命令）。该命令的输出将是指示成功、失败或错误的代码，或者是 32 字节的 MAC。在运行该命令之前，应使用 Nonce 命令在固定模式下将报文写入到 TempKey 或报文摘要缓冲区。在该模式下，器件仅加速公钥计算并返回布尔结果。

通过外部公钥验证报文的步骤

1. 使用 Nonce 命令在固定模式下将 32 字节的报文摘要写入 TempKey 或报文摘要缓冲区。
2. 可选：系统 Nonce——系统生成的 nonce。
   2.1. 如果外部报文摘要存储在 TempKey 中，则系统生成的 nonce 必须存储在报文摘要缓冲区的低 32 个字节中。
   2.2. 如果外部报文存储在 MessageDigestBuffer[31:0]中，则系统 Nonce 必须存储在报文摘要缓冲区的高 32 个字节[63:32]中。为此，应将外部报文和 nonce 值组合在一起以 64 字节值的形式写入。
3. 发出 Verify 命令。其中包括模式、KeyID（指定 P256 ECC 曲线）、64 字节签名和 64 字节外部公钥。
4. 输出将返回：
   4.1. 一个字节，即成功、失败或错误代码（不需要 MAC 时）。
   4.2. 一个 32 字节 MAC（如果模式指定）。

表 5-51. 命令参数

操作码（1 字节）	模式（1 字节）	Key ID（2 字节）	数据字段（128 字节）		注释
			签名（64 字节）	公钥（64 字节）	
0x45	0x02	0x00 04	R 值 S 值	X 值 Y 值	报文存储在 TempKey 中
	0x22	0x00 04	R 值 S 值	X 值 Y 值	报文存储在报文摘要缓冲区中
	0xA2	0x00 04	R 值 S 值	X 值 Y 值	• 报文存储在 TempKey 中 • 系统 Nonce 存储在 MDB[31:0]中 • 返回生效 MAC
	0x82	0x00 04	R 值 S 值	X 值 Y 值	• 报文存储在报文摘要缓冲区中 • 系统 Nonce 存储在 MDB[63:32]中 • 返回生效 MAC

表 5-52. 输出响应——外部验证

名称	模式	大小	响应
Response	0x02 或 0x22	1 字节	• 0x00——签名通过验证时 • 0x01——签名不匹配时 • 错误代码——由于某些其他原因导致失败时
	0x82 或 0xA2	1 字节或 32 字节	• 生效 MAC——签名通过验证时 • 0x01——签名不匹配时 • 错误代码——由于某些其他原因导致失败时

表 5-53. 生效 MAC——外部验证

大小（字节）	报文存储在 TempKey 中	报文存储在报文摘要缓冲区中
32	IO 保护密钥的内容	IO 保护密钥的内容
32	报文存储在 TempKey 中	报文存储在报文摘要缓冲区的前 32 个字节中
32	系统 Nonce 存储在报文摘要缓冲区的前 32 个字节中	系统 Nonce 存储在报文摘要缓冲区的后 32 个字节中
32	所传递签名的 R 数据	所传递签名的 R 数据
32	所传递签名的 S 数据	所传递签名的 S 数据
1	操作码	操作码
1	模式	模式
2	Param2 [LSB,MSB]	Param2 [LSB,MSB]

5.2.4.2 验证——存储公钥模式

在存储模式下使用 `Verify` 命令时，要使用的公钥存储在数据槽中，无需传递。在运行此命令之前，应使用 `Nonce` 命令将报文写入 TempKey 或报文摘要缓冲区。

通过存储的公钥验证报文的步骤

1. 必要时，在使用 Verify 命令之前先使公钥生效。
2. 如有需要，请先授权再使用公钥。
3. 使用 Nonce 命令在固定模式下将 32 字节的报文摘要写入 TempKey 或报文摘要缓冲区。
4. 将系统 Nonce 写入报文摘要缓冲区的低 32 个字节或高 32 个字节。
   4.1. 如果 TempKey 包含报文摘要，则将系统 Nonce 存储在报文摘要缓冲区的低 32 个字节中。
   4.2. 如果报文摘要存储在报文摘要缓冲区的低 32 个字节中，则将系统 Nonce 存储在报文摘要缓冲区的高 32 个字节中。用户需要使用 Nonce 命令一次性写入报文摘要和系统 Nonce。
5. 发出 Verify 命令。其中包括模式、KeyID（指定公钥槽）、64 字节签名和内部公钥的槽号。
6. 输出将返回：
   6.1. 一个字节，即成功、失败或错误代码（不需要 MAC 时）。
   6.2. 如果命令失败，则为一个 32 字节 MAC（如果模式或错误代码指定）。

表 5-54. 命令参数

操作码（1 字节）	模式（1 字节）	公钥（2 字节）	数据字段（64 字节）	注释
			签名（64 字节）	
0x45	0x00	<0x00, 0[槽号]	R 值 S 值	- 报文存储在 TempKey 中
	0x20	<0x00, 0[槽号]	R 值 S 值	- 报文存储在报文摘要缓冲区中
	0x80	<0x00, 0[槽号]	R 值 S 值	- 报文存储在 TempKey 中 返回生效 MAC
	0xA0	<0x00, 0[槽号]	R 值 S 值	- 报文存储在报文摘要缓冲区中 返回生效 MAC

表 5-55. 输出响应——存储验证

名称	模式	大小	响应
Response	0x00 或 0x20	1 字节	• 0x00——签名通过验证时 • 0x01——签名不匹配时 • 错误代码——由于某些其他原因导致失败时
	0x80 或 0xA0	32 字节 1 字节	• 生效 MAC——签名通过验证时 • 0x01——签名不匹配时 • 错误代码——由于某些其他原因导致失败时

表 5-56. 生效 MAC——存储验证

生效 MAC 输入位置取决于报文的存储位置。

大小（字节）	报文存储在 TempKey 中	报文存储在报文摘要缓冲区中
32	IO 保护密钥的内容	IO 保护密钥的内容
32	TempKey 中的报文	摘要缓冲区中的前 32 个字节报文
32	报文摘要缓冲区的前 32 个字节中存储的系统 Nonce	报文摘要缓冲区的后 32 个字节中存储的系统 Nonce
32	所传递签名的 R 数据	所传递签名的 R 数据

.....（续）		
大小（字节）	报文存储在 TempKey 中	报文存储在报文摘要缓冲区中
32	所传递签名的 S 数据	所传递签名的 S 数据
1	操作码	操作码
1	模式	模式
2	Param2 [LSB,MSB]	Param2 [LSB,MSB]

5.3 对称加密命令

5.3.1 AES 命令

AES 命令可用于利用 AES 密钥对 16 字节的数据块进行加密和/或解密。请注意，密钥存储在给定槽的 16 字节（128 位）存储单元内或 TempKey 的前 16 个字节内。可将多个密钥存储在一个给定槽中，并按 16 字节边界连续访问（从字节 0-15 开始，一直到该槽的最后 16 个字节），但任何槽中的密钥数量都不超过四个。对于 ATECC608A-TNGTLS，AES 密钥可存储在 Slot 9 中。Slot 9 最多可容纳 4 个 AES 密钥。

除了 AES 加密和解密外，AES 命令还可用于生成 Galois 域乘法（Galois Field Multiply，GFM），以支持其他加密操作。

5.3.1.1 AES-ECB 加密

在 AES-ECB 加密模式下，输入流中需为 16 字节明文，器件的输出将为 16 字节密文。

表 5-57. AES-ECB 加密

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据（16 字节）	注释
0x51	0x00 0x40 0x80 0xC0	0x00 0[槽号]	任意 16 字节的明文数据	在位置 0 加密密钥 在位置 1 加密密钥 在位置 2 加密密钥 在位置 3 加密密钥
	0x00	0xFF FF	任意 16 字节的明文数据	加密密钥位于 TempKey 中

表 5-58. 输出响应——AES 加密

名称	大小（字节）	注
Response	1	如果操作失败，则输出为一个 1 字节错误代码。
	16	如果操作成功，器件将输出 16 字节的密文。

5.3.1.2 AES-ECB 解密

AES 命令的 AES-ECB 解密模式用于将密文转换回明文。

表 5-59. AES-ECB 解密

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据（16 字节）	注释
0x51	0x01 0x41 0x81 0xC1	0x00 0[槽号]	任意 16 字节的 AES 加密数据	在位置 0 解密密钥 在位置 1 解密密钥 在位置 2 解密密钥 在位置 3 解密密钥
	0x01	0xFF FF	任意 16 字节的明文数据	解密密钥位于 TempKey 中

表 5-60. 输出响应——AES 解密

名称	大小（字节）	注
Response	1	如果操作失败，则输出为一个 1 字节错误代码。
	16	如果操作成功，器件将输出 16 字节的明文数据。

5.3.1.3 AES-GFM

GFM 操作通常用作各种 AES 加密操作的一部分。该功能旨在帮助创建 ATECC608A-TNGTLS 不直接支持的操作。该操作的输出可用于 AES-GCM AEAD 功能。该模式不涉及机密信息或芯片上存储的任何内容。如果选择该模式，则其余模式位将被忽略。

表 5-61. AES Galois 域乘法（GFM）

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据（32 字节）	说明
0x51	0x03	0x00 00	• 前 16 个字节——磁场 • 后 16 个字节——输入数据	

表 5-62. 输出响应——AES GFM

名称	大小（字节）	注
Response	1	如果操作失败，则输出为一个 1 字节错误代码。
	16	如果操作成功，器件将输出 16 字节的 GFM 计算结果。

5.3.2 CheckMac 命令

CheckMac 命令可计算不同 CryptoAuthentication™（ATECC608A、ATECC508A 和 ATSHA204A）器件上生成的 MAC 响应，然后将结果与输入值进行比较。该命令会返回一个布尔结果来指示比较成功还是失败。

如果 TempKey 中的值用作 CheckMac 的输入，则 Nonce 和/或 GenDig 命令必须在 CheckMac 命令之前运行。

表 5-63. 输入参数——CheckMac

操作码（1 字节）	模式（1 字节）(2)	KeyID（2 字节）	数据（77 字节）(1)	说明
0x28	0x00	0x00 0[槽号]	• 32 字节客户端质询 • 客户端生成的 32 字节响应 • 13 字节其他数据	
	0x01	0x00 0[槽号]	• 32 字节被忽略，但必须存在 • 32 字节客户端响应 • 13 字节其他数据	TempKey.SourceFlag 随机时使用
	0x05	0x00 0[槽号]		TempKey.SourceFlag 固定时使用
	0x02	0x00 00	• 32 字节客户端质询 • 32 字节客户端响应 • 13 字节其他数据	TempKey.SourceFlag 随机时使用
	0x06	0x00 00		TempKey.SourceFlag 固定时使用

注：

1. OtherData[0:12]值必须与原始 MAC 命令中使用的值匹配。
2. 对于 0x00 以外的模式，Mode[2]必须与 TempKey.SourceFlag 匹配。

表 5-64. 输出响应——CheckMac

名称	大小	说明
Response	1 字节	• 0x00——表示成功 • 0x01——表示存在不匹配 • 错误代码——表示失败

表 5-65. SHA256 CheckMac 哈希运算

字节数	模式 0x00	模式 0x01 或 0x05	模式 0x02 或 0x06
32	Key[KeyID]	Key[KeyID]	TempKey
32	输入客户端质询	TempKey	输入客户端质询
4	OtherData[0:3]	OtherData[0:3]	OtherData[0:3]
8	0	0	0
3	OtherData[4:6]	OtherData[4:6]	OtherData[4:6]
1	SN[8] = 0x01	SN[8] = 0x01	SN[8] = 0x01
4	OtherData[7:10]	OtherData[7:10]	OtherData[7:10]
2	SN[0:1] = 0x01 0x23	SN[0:1] = 0x01 0x23	SN[0:1] = 0x01 0x23
2	OtherData[11:12]	OtherData[11:12]	OtherData[11:12]

5.3.3 GenDig 命令

GenDig 命令使用 SHA-256 哈希运算将存储的或输入的值与 TempKey 的内容相结合，这些内容在执行此命令之前必须有效。存储的值可以来自其中一个数据槽、配置区域、任一 OTP 页或单调计数器。器件的具体模式将确定 GenDig 计算中要包含的数据。

在某些情况下，在执行某些命令之前，需要先运行 GenDig。该命令可运行多次，以在执行给定命令之前在摘要中包含更多数据。得到的摘要保留在 TempKey 中，可以通过以下四种方式之一使用：

1. 它可以作为 MAC、Sign 或 CheckMac 命令使用的报文的一部分包含在内。由于 MAC 响应输出结合了 GenDig 计算中使用的数据与 MAC 命令的机密信息密钥，因此将用于验证数据和/或 OTP 区域中存储的数据。
2. 后续读或写命令可以使用摘要来为数据提供认证和/或加密，在这种情况下，相应摘要称为数据保护摘要。
3. 可以使用此命令通过传输密钥阵列中的值进行安全个性化。得到的数据保护摘要将供写入操作使用。
4. 输入值（通常是远程器件中的 nonce）与当前 TempKey 值相结合以创建一个共用的 nonce，其中两个器件均可证明包含 RNG。

5.3.3.1 GenDig——配置

GenDig 计算中可以包括来自配置区域的数据。数据始终包含在 32 字节的块中，并且任何给定的 GenDig 计算中只能包含一个块。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。后续 GenDig 命令将使用上一 GenDig 操作中存储在 TempKey 中的值。

表 5-66. 输入参数——GenDig 配置

操作码（1 字节）	模式（1 字节）	KeyID[(1)]（2 字节）	数据（0 字节）	说明
0x15	0x00	0x00 00	—	使用配置块 0
		0x00 01	—	使用配置块 1
		0x00 02	—	使用配置块 2
		0x00 03	—	使用配置块 3

注：

1. KeyId 指定要在 TempKey 计算中使用的配置区域块。

表 5-67. 输出响应——GenDig 配置

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将设置为 0。

表 5-68. TempKey 计算——GenDig 配置

32 字节	配置区域块
1 字节	操作码 = 0x15
1 字节	模式 = 0x00
2 字节	KeyID[0:1] = 0x0[block] 0x00
1 字节	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23
25 字节	全零
32 字节	TempKey.Value

5.3.3.2 GenDig——OTP

GenDig 计算中可以包括来自 OTP 区域的数据。数据始终包含在 32 字节的块中，并且任何给定的 GenDig 计算中只能包含一个块。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。后续 GenDig 命令将使用上一 GenDig 操作中的值。

表 5-69. 输入参数——GenDig OTP

操作码（1 字节）	模式（1 字节）	KeyID[1]（2 字节）	数据（0 字节）	说明
0x15	0x01	0x00 00	—	使用 OTP 块 0 作为 KeyID
		0x00 01	—	使用 OTP 块 1 作为 KeyID

注：

1. KeyId 指定要在 TempKey 计算中使用的 OTP 区域块。

表 5-70. 输出响应——GenDig OTP

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将设置为 0。

表 5-71. TempKey 计算——GenDig OTP

32 字节	OTP 区域块
1 字节	操作码 = 0x15
1 字节	模式 = 0x00
2 字节	KeyID[0:1] = 0x0[block] 0x00
1 字节	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23
25 字节	全零
32 字节	TempKey.Value

5.3.3.3 GenDig——数据

GenDig 计算中可以包括来自数据区域槽的数据。数据始终包含在 32 字节的块中，GenDig 计算中只能包含槽的低字节块。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。如果槽需要随机 nonce，则数据必须由 Nonce 命令生成，而非传递给器件。后续 GenDig 命令将使用上一 GenDig 操作中的值。

如果为 NoMAC 设置了槽，则不允许将此 GenDig 的输出用于 MAC 命令。运行多个 GenDig 时，如果任何槽设置了 NoMAC，则 TempKey 中的输出无法用于执行 MAC 命令。

表 5-72. 输入参数——GenDig 数据

操作码（1 字节）	模式（1 字节）	KeyID[1]（2 字节）	数据（0 或 4 字节）	说明
0x15	0x02	0x00 0[槽号]	—	计算中使用来自槽 #的数据。
		0x00 0[槽号]	4 字节	计算中使用来自槽 #的数据以及其他数据（使用 NoMAC 密钥槽时）

注：

1. KeyId 指定要在 TempKey 计算中使用的数据区域槽。仅使用低 32 个字节。

表 5-73. 输出响应——GenDig 数据

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将置 1，表示在计算中使用了数据区域槽。
3. TempKey.KeyID 将设置为命令中指定的槽。
4. 如果允许 MAC 命令，则 TempKey.NoMacFlag 将设置为 0；如果不允许则置 1。

表 5-74. TempKey 计算——GenDig 数据

大小	参数——MAC	参数——NoMAC
32 字节	DATA.slot[KeyID]	DATA.slot[KeyID]
4 字节	操作码、模式和 KeyID = 0x15、0x02 和 0x0[槽号] 0x00	OtherData[0:3]
1 字节	SN[8] = 0x01	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23	SN[0:1] = 0x01 0x23
25 字节	全零	全零
32 字节	TempKey.Value	TempKey.Value

5.3.3.4 GenDig——共用 Nonce

在共用 Nonce 模式下，该命令的输入为 32 字节数据。当两个器件必须共用 nonce 时，使用该模式。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。后续 GenDig 命令将使用上一 GenDig 操作中的值。

表 5-75. 输入参数——GenDig 共用 Nonce

操作码 （1 字节）	模式 （1 字节）	KeyID(1) （2 字节）	数据 （32 字节）	说明
0x15	0x03	0x00 0[槽号]	输入数据	
		0x80 0[槽号]	输入数据	

注：

1. KeyId 指定要在 TempKey 计算中使用的数据区域槽。仅使用低 32 个字节。

表 5-76. 输出响应——GenDig 数据

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将置 1，表示在计算中使用了数据区域槽。
3. TempKey.KeyID 将设置为命令中指定的槽。

表 5-77. TempKey 计算——GenDig 共用 Nonce

大小	参数——KeyID MSB 0x00	参数——KeyID MSB 0x80
32 字节	输入数据	TempKey.value
1 字节	操作码 = 0x15	操作码 = 0x15
1 字节	模式 = 0x03	模式 = 0x03
1 字节	KeyID 的 LSB = 0x0[槽号]	KeyID 的 LSB = 0x0[槽号]
1 字节	0x00	0x00

.....（续）		
大小	参数——KeyID MSB 0x00	参数——KeyID MSB 0x80
1 字节	SN[8] = 0x01	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23	SN[0:1] = 0x01 0x23
25 字节	全零	全零
32 字节	TempKey.Value	输入数据

5.3.3.5 GenDig——计数器

在 GenDig 命令的计数器模式下，计数器的二进制值包含在 TempKey 计算中。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。后续 GenDig 命令将使用上一 GenDig 操作中的值。

表 5-78. 输入参数——GenDig 计数器

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据（0 字节）	说明
0x15	0x04	0x00 00	—	包含 Counter[0]值
		0x00 01	—	包含 Counter[1]值

注：

1. KeyId 指定要在 TempKey 计算中使用的单调计数器值。

表 5-79. 输出响应——GenDig 计数器

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将设置为 0。

表 5-80. TempKey 计算——GenDig 计数器

32 字节	全零
1 字节	操作码 = 0x15
1 字节	模式 = 0x04
2 字节	KeyID[0:1] = 0x0[Counter#] 0x00
1 字节	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23
1 字节	零
4 字节	Counter[KEYID]——Counter 命令报告的二进制值
20 字节	全零
32 字节	TempKey.Value

5.3.3.6 GenDig——密钥配置

在 GenDig 命令的密钥配置模式下，GenDig TempKey 计算中包含由 KeyID 指定的密钥的槽配置和密钥配置。Nonce 命令必须在第一个 GenDig 命令之前运行，以将值装入 TempKey 中。后续 GenDig 命令将使用上一 GenDig 操作中的值。

表 5-81. 输入参数——GenDig 密钥配置

操作码（1 字节）	模式（1 字节）	KeyID[(1)]（2 字节）	数据（0 或 4 字节）	说明
0x15	0x05	0x00 0[槽号]	—	包含槽配置信息

注：

1. KeyId 指定将在 TempKey 计算中包含其配置信息的槽号。不包含实际的槽值。

表 5-82. 输出响应——GenDig 密钥配置

名称	大小	说明
Response	1 字节	0x00——表示成功。 错误代码——表示命令失败。

注： 标志位

1. 如果成功，TempKey.Valid 标志将置 1，否则将为 0。
2. TempKey.GenDigData 将设置为 0。

表 5-83. TempKey 计算——GenDig 密钥配置

32 字节	TempKey.value
1 字节	操作码 = 0x15
1 字节	模式 = 0x05
2 字节	KeyID[0:1] = 0x0[槽号] 0x00
1 字节	SN[8] = 0x01
2 字节	SN[0:1] = 0x01 0x23
1 字节	0x00
2 字节	SlotConfig[KeyID]
2 字节	KeyConfig[KeyID]
1 字节	SlotLocked[KeyID]
20 字节	全零

5.3.4 KDF 命令

KDF 命令实现了多个密钥派生函数（Key Derivation Function，KDF）。通常，该函数将源密钥与输入字符串组合在一起，并创建对称的结果密钥/摘要/数组。输入密钥可位于 TempKey、备用密钥缓冲区或 EEPROM 槽中，并且长度可为 32 字节或 64 字节。

KDF 的输出结果（可能为 32 字节或 64 字节）可以在输出缓冲区中返回给系统、写入 TempKey 或备用密钥缓冲区或者存储在 EEPROM 槽中。仅当低 32 个字节区域已经有效时，才可以将 32 字节 KDF 结果写入 TempKey 的高 32 个字节区域。

5.3.4.1 KDF——PRF

PRF 是 TLS 1.2 及以下版本中指定的密钥派生函数，用于建立会话。该芯片支持多种变体，包括用于生成主机密信息、使会话（完成的报文）生效和生成密钥材料（包括 AEAD 套件）的方法。

表 5-84. 输入参数——KDF PRF

操作码（1 字节）	模式（1 字节）	KeyId[1]（2 字节）	详细信息（4 字节）	数据（0 至 127 字节）	说明	
					输入源	输出结果
0x56	0x00 0x04 0x08 0x0C 0x10 0x14	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-85	通过表 5-85 中的 DataLen 指定	TempKey（32 或 64 字节）	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x01 0x05 0x09 0x0D 0x11 0x15	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-85	通过表 5-85 中的 DataLen 指定	临时密钥高字节块——32 字节	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x02 0x06 0x0A 0x0E 0x12 0x16	0x00 0[SI] 0x00 0[SI] 0x0[SO] 0[SI] 0x00 0[SI] 0x00 0[SI] 0x00 0[SI]	请参见表 5-85	通过表 5-85 中的 DataLen 指定	EEPROM 槽	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x03 0x07 0x0B 0x0F 0x13 0x17	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-85	通过表 5-85 中的 DataLen 指定	备用密钥缓冲区	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文

注：

1. [SO] = 输出槽号，[SI] = 输入槽号。

表 5-85. PRF 的详细参数编码

Bit	名称	说明
31-24	DataLen	输入参数的长度（字节）（在 TLS 语言中为 Label | Seed）。
23-11	Zero	所有位必须为 0。
10-9	Aead	00 = 不进行特殊的 AEAD 处理。 01 = TargetLen 必须为 1 才能生成 64 个字节。 • 前 32 个字节进入目标位置，不得输出 • 后 32 个字节进入输出缓冲区（始终不加密） 禁止使用 10 和 11，否则将返回解析错误。
8	TargetLen	要置于目标位置的 32 字节块的数量。 0 = 1 个块（32 字节） 1 = 2 个块（64 字节）
7-2	Zero	所有位必须为 0。

.....（续）

Bit	名称	说明
1-0	KeyLen	源密钥的长度（以 16 字节块为单位）。 0 = 1 个块（16 字节） 1 = 2 个块（32 字节） 2 = 3 个块（48 字节） 3 = 4 个块（64 字节）

表 5-86. 输出响应——KDF PRF

名称	模式	大小	说明
OutData	0x00 至 0x0F	1	输出发送到内部存储单元的所有模式的成功或失败代码。
	0x10 至 0x17	32 或 64	输出置于输出缓冲区中的模式的输出数据。
OutNonce	0x00 至 0x13	0	如果输出未加密。
	0x14 至 0x17	32	如果输出已加密，则将生成并输出随机 nonce。

5.3.4.2 KDF——AES

KDF 命令的 AES 模式在单个输入数据块上计算 AES-ECB。输出的高 16 字节均填充 0，因此结果始终为 32 字节。

表 5-87. 输入参数——KDF AES

操作码（1 字节）	模式（1 字节）	KeyId(1)（2 字节）	详细信息（4 字节）	数据（16 字节）	说明	
					输入密钥源（16 字节）	输出结果
0x56	0x20 0x24 0x28 0x2C 0x30 0x34	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-88	要加密的数据	TempKey（32 或 64 字节）	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x21 0x25 0x29 0x2D 0x31 0x35	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-88	要加密的数据	临时密钥高字节块——32 字节	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x22 0x26 0x2A 0x2E 0x32 0x36	0x00 0[SI] 0x00 0[SI] 0x0[SO] 0[SI] 0x00 0[SI] 0x00 0[SI] 0x00 0[SI]	请参见表 5-88	要加密的数据	EEPROM 槽	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x23 0x27 0x2B 0x2F 0x33 0x37	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-88	要加密的数据	备用密钥缓冲区	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文

注：

1. [SO] = 输出槽号，[SI] = 输入槽号。

表 5-88. AES 的详细参数编码

Bit	名称	说明
31-2	Zero	所有位必须为 0。
1-0	KeyLoc	AES 密钥位于源密钥材料中的 Src[KeyLoc*16]。 • 对于 64 字节存储单元，密钥可以位于 4 个块之一中。 • 对于 32 字节存储单元，密钥可以位于块 0 或块 1 中。

表 5-89. 输出响应——KDF AES

名称	模式	大小	说明
OutData	0x20 至 0x2F	1	输出发送到内部存储单元的所有模式的成功或失败代码。
	0x30 至 0x37	32	输出置于输出缓冲区中的模式的输出数据。
OutNonce	0x20 至 0x33	0	如果输出未加密。

.....（续）

名称	模式	大小	说明
	0x34 至 0x37	32	如果输出已加密，则将生成并输出随机 nonce。

5.3.4.3 KDF——HKDF

KDF 命令中的 HKDF 函数旨在支持 TLS 1.3 和其他协议中指定的必要密钥派生运算。该函数始终使用模式和详细信息参数中指定的密钥和报文来计算 HMACSHA256 的单次迭代。根据 HKDF 规范，可使用该命令的多次迭代实现 HKDF 提取和扩展函数。通过分别写入 TempKey 寄存器的上半部分和下半部分，可以创建 64 字节的结果。

对于 ATECC608A-TNGTLS，特殊 IV 功能已在配置区域中禁止，因此无法使用。

表 5-90. 输入参数——KDF HKDF

操作码（1 字节）	模式（1 字节）	KeyId[2]（2 字节）	详细信息（4 字节）	数据（0 至 127 字节）	说明	
					输入源	输出结果
0x56	0x00[1] 0x44 0x48 0x4C 0x50 0x54	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-91	通过表 5-91 中的 DataLen 指定	TempKey（32 或 64 字节）	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x41 0x45 0x49 0x4D 0x51 0x55	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-91	通过表 5-91 中的 DataLen 指定	临时密钥高字节块——32 字节	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x42 0x46 0x4A 0x4E 0x52 0x56	0x00 0[SI] 0x00 0[SI] 0x0[SO] 0[SI] 0x00 0[SI] 0x00 0[SI] 0x00 0[SI]	请参见表 5-91	通过表 5-91 中的 DataLen 指定	EEPROM 槽	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文
	0x43 0x47 0x4B 0x4F 0x53 0x57	0x00 00 0x00 00 0x0[SO] 00 0x00 00 0x00 00 0x00 00	请参见表 5-91	通过表 5-91 中的 DataLen 指定	备用密钥缓冲区	TempKey 临时密钥——高字节块 EEPROM 槽 备用密钥缓冲区 OutBuf——明文 OutBuf——密文

⚠ CAUTION

1. 如果输出和输入均指向 TempKey 的低 32 字节，则必须将详细信息参数的 MsgLoc 部分设置为 TempKey（MsgLoc = 01）。如果发生这种情况，结果将不可预测。禁止这种组合。
2. [SO] = 输出槽号，[SI] = 输入槽号。

表 5-91. HKDF 的详细参数编码

Bit	名称	说明
31-25	DataLen	HKDF 报文的长度（以字节为单位）。如果该值为零，则报文将是 32 字节的 0x00。
24-12	Zero	所有位必须为 0。
11-8	msgKey	报文的密钥槽（位于 EEPROM 中时）。
7-3	Zero	所有位必须为 0。
2	ZeroKey	如果为 1，则密钥为 32 字节的 0x00。
1-0	MsgLoc	报文的位置。 00 = EEPROM 槽 01 = TempKey 10 = 输入参数 11 = 已针对 ATECC608A-TNGTLS 禁止。

表 5-92. 输出响应——KDF HKDF

名称	模式	大小	说明
OutData	0x40 至 0x4F	1	输出发送到内部存储单元的所有模式的成功或失败代码。
	0x50 至 0x57	32 或 64	输出置于输出缓冲区中的模式的输出数据。
OutNonce	0x40 至 0x53	0	如果输出未加密。
	0x54 至 0x57	32	如果输出已加密，则将生成并输出随机 nonce。

5.3.4.4 KDF 输出加密

对于所有 KDF 算法，均已采用类似于 ECDH 命令的方式实现输出缓冲区加密。尽管不需要加密，但是加密有助于额外提高系统安全性。对于 ATECC608A-TNGTLS，已使能输出缓冲区加密。用于加密的 IO 保护密钥已存储在 Slot 6 中。请参见每种工作模式来确定具体有哪些模式值需要加密。

以下过程用于生成加密输出：

1. IO 保护密钥槽的前 32 个字节（Config.ChipOptions[15:12]）复制到 SHA256 缓冲区。
2. 内部 RNG 生成一个 32 字节的随机数，并将该 nonce 的前 16 个字节附加到 SHA256 缓冲区
3. SHA256 缓冲区进行哈希运算，摘要与明文 KDF 结果的前 32 个字节进行异或运算。如果结果中只有 16 个字节，则输出缓冲区将仅包含这 16 个字节，SHA 摘要的后 16 个字节将被忽略。
4. 如果输出中超过 32 个字节，则将通过对 IO 保护密钥（32 个字节）进行 SHA256 哈希运算，然后在哈希运算结果后附加步骤 2 中随机 nonce 的后 16 个字节来创建新摘要。最终得到的摘要将与结果的后 32 个字节进行异或运算。
5. 输出缓冲区包含加密的 KDF 结果及其后的 32 字节 nonce。即使仅使用了 nonce 的前 16 个字节，也会输出 nonce 的全部 32 个字节。

5.3.5 MAC 命令

报文验证代码（MAC）命令用于生成报文的 SHA256 摘要，摘要由器件中存储的密钥、质询和器件上的其他信息组成。此命令的输出为此报文的摘要。

使用此命令的正常命令流程如下所示：

1. 运行 Nonce 命令以加载输入质询，并视情况将其与生成的随机数组合。此操作的结果是在器件内部存储的一个 nonce。

2. 视情况运行一次或多次 GenDig 命令，将器件中的存储 EEPROM 单元与 nonce 组合。结果存储在器件内部。此功能允许将 2 个或多个密钥用作响应生成的一部分。
3. 运行此 MAC 命令，将步骤 1（以及步骤 2，需要时）的输出与 EEPROM 密钥组合，以生成一个输出响应（即，摘要）。

也可通过相同的 GenDig 机制将任何槽中的数据（并非必须保密）都累积到响应中。其效果是验证相应单元中存储的值。

5.3.5.1 非多样化 MAC

MAC 始终基于全部 88 个字节来计算，并且始终创建一个 32 字节的 SHA256 摘要。非多样化 MAC 不包括器件的序列号，因此，如果输入参数相同，则所有器件的 MAC 都将相同。

表 5-93. 输入参数——非多样化 MAC

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据(2)（0-32 字节）	模式说明
0x08	0x00	0x00 0[槽号]	32 字节	• 前 32 个字节从数据槽装入 • 后 32 个字节取自输入质询
	0x01 或 0x05(1)	0x00 0[槽号]	0 字节	• 前 32 个字节从数据槽装入 • 后 32 个字节取自 TempKey
	0x02 或 0x06(1)	0x00 00	32 字节	• 前 32 个字节从 TempKey 装入 • 后 32 个字节取自输入质询

注：

(1)模式[2]必须与 TempKey.SourceFlag 匹配。
(2)如果存在，则数据参数对应于输入质询。

表 5-94. 输出响应——非多样化 MAC

名称	大小	说明
Response	1 字节	如果命令失败
	32 字节	SHA-256 摘要

表 5-95. 非多样化 MAC 计算

字节数	模式 0x00	模式 0x01 或 0x05	模式 0x02 或 0x06
32 32 1 1 2 11 1 4 2 2	数据槽 输入质询 操作码（0x08） 模式 KeyID 0 SN[8] 0x01 0 SN[0:1] 0x01 0x23 0	数据槽 TempKey 操作码（0x08） 模式 KeyID 0 SN[8] 0x01 0 SN[0:1] 0x01 0x23 0	TempKey 输入质询 操作码（0x08） 模式 KeyID 0 SN[8] 0x01 0 SN[0:1] 0x01 0x23 0

5.3.5.2 多样化 MAC

多样化 MAC 包括器件的序列号。每个器件的序列号都是惟一的，因此将始终生成惟一的 SHA256 摘要。MAC 始终基于全部 88 个字节来计算，并且始终创建一个 32 字节的 SHA256 摘要。

表 5-96. 输入参数——多样化 MAC

操作码（1 字节）	模式（1 字节）	KeyID（2 字节）	数据(2)（0-32 字节）	模式说明
0x08	0x40	0x00 0[槽号]	32 字节	• 前 32 个字节从数据槽装入 • 后 32 个字节取自输入质询
	0x41 或 0x45(1)	0x00 0[槽号]	0 字节	• 前 32 个字节从数据槽装入 • 后 32 个字节取自 TempKey
	0x42 或 0x46(1)	0x00 00	32 字节	• 前 32 个字节从 TempKey 装入 • 后 32 个字节取自输入质询

注：

(1)模式[2]必须与 TempKey.SourceFlag 匹配。
(2)如果存在，则数据参数对应于输入质询。

表 5-97. 输出响应——多样化 MAC

名称	大小	说明
Response	1 字节	如果命令失败
	32 字节	SHA-256 摘要

表 5-98. 多样化 MAC 计算

字节数	模式 0x40	模式 0x41 或 0x45	模式 0x42 或 0x46
32 32 1 1 2 11 1 4 2 2	数据槽 输入质询 操作码（0x08） 模式 KeyID 0 SN[8]0x01 SN[4:7] SN[0:1] 0x01 0x23 SN[2:3]	数据槽 TempKey 操作码（0x08） 模式 KeyID 0 SN[8] 0x01 SN[4:7] SN[0:1] 0x01 0x23 SN[2:3]	TempKey 输入质询 操作码（0x08） 模式 KeyID 0 SN[8] 0x01 SN[4:7] SN[0:1] 0x01 0x23 SN[2:3]

6. 应用信息

ATECC608A-TNGTLS 是 Microchip Trust＆GO CryptoAuthentication™产品系列的成员。Trust&GO 产品不但使用方便，而且易于实现，甚至允许小规模生产的客户在其最终系统中实现安全性，同时利用 Microchip 的专业知识和基础架构进行安全配置。

ATECC608A-TNGTLS 器件旨在为 IoT 联网产品轻松增加安全性。该产品已经过预先配置，可以通过 TLS 连接轻松连接到 IoT 云，并能够为其他多个安全用例提供支持，包括安全引导、一次性用品和配件认证以及用户数据和 I/P 保护。

除了实际的安全器件之外，Microchip 还开发了一系列工具，这些工具可与我们的硬件器件无缝集成，从而为开发整个安全解决方案提供便捷途径。当开发人员使用 Microchip 的软件安全工具时，可消除自行设置基础架构的复杂性，并且能够快速进行初始原型设计和生产。

6.1 用例

ATECC608A-TNGTLS 专门面向物联网市场而打造。该器件已设置为直接支持 Google IoT Cloud™。通过向 Slot 8 添加证书，也可支持 Amazon AWS®或其他 TLS 服务器。该器件还可以支持其他用例。下面简要介绍了该器件的主要用例。

安全的 TLS 连接
ATECC608A-TNGTLS 允许使用多种协议创建安全的 TLS 连接。该器件能够与 Google Cloud、AWS 和其他云提供商建立安全连接。通过不同模式的密钥派生函数（KDF），可以生成适当的密钥来支持 TLS1.2、TLS1.3 和更早的安全连接 Internet 协议。

6.2 开发工具

ATECC608A-TNGTLS 配有多种软硬件支持工具以及后端服务，便于快速开发应用程序。初始开发阶段可以借助易于使用的 Trust Platform Design Suite 工具系列。该系列工具提供了一种图形化的方式来实现您的用例，并且最终会生成您的应用程序所需的 C 代码。

如果预定义的 Trust Platform Design Suite 工具无法实现您的应用程序，则可以使用 CryptoAuthLib 或 Python®版本的 CryptoAuthLib 和 CryptoAuthTools 来开发应用程序。CryptoAuthLib 也是从 Trust Platform Design Suite 工具输出的代码主干。

可以通过硬件工具以及 ATECC608A-TNGTLS 器件示例来对您的应用程序进行全面验证。由于已设置器件的访问策略，因此重点围绕系统级代码开发进行验证。

实现应用程序之后，可以通过 Microchip 直销网站订购 ATECC608A-TNGTLS 器件。

6.2.1 Trust Platform Design Suite

为了简化实现过程，Microchip 开发了基于 Web 的 Trust Platform Design Suite 工具，这套工具可通过流程引导开发人员了解从概念到生产的整个过程。使用这套工具，您可以遵循 ATECC608A-TNGTLS 的配置及指定访问策略的约束来开发和构建实现特定应用程序所需的事务图和代码。

有关这些工具的更多信息，请参见网页“安全 IC”部分下的“Microchip CryptoAuthentication 产品”。

6.2.2 硬件工具

使用 ATECC608A-TNGTLS 进行开发时，有多种硬件工具可为您提供帮助。关于此处未提及的其他工具的可用性信息，请访问 Microchip 网站。工具相关信息中通常也会提供用例示例。

DM320118——CryptoAuthentication Trust Platform
DM320118 是一套紧凑型开发系统，其中包含 ATSAMD21 单片机，ATECC608A-TNGTLS、ATECC608A-TFLXTLS 和 ATECC608-TCSTM Trust 器件各 1 个，USB 集线器，mikroBUS 连接器和板上调试器。该工具包旨在与 Trust

Platform Design Suite 工具搭配使用，后者用于实现 ATECC608A-TFLXTLS 器件的各种用例。该工具包可与 MPLAB X 或 Atmel Studio Design 环境搭配使用，以开发其他应用程序。

DM320109——CryptoAuthentication 入门工具包
DM320109 由 ATSAMD21-XPRO 开发板组成，该开发板预编程了可以与 CryptoAuthentication 器件搭配使用的固件。该工具包随附 AT88CKSCKTSOIC-XPRO 插座板，但您需要获取该插座板的 UDFN 版本才能与当前仅以 UDFN 封装提供的示例器件搭配使用。ATECC608A-TNGTLS 的特定示例将需要单独获取。

AT88CKSCKTUDFN(SOIC)-XPRO
AT88CKSCKTUDFN-XPRO 和 AT88CKSCKTSOIC-XPRO 是通用的 CryptoAuthentication 插座工具包，可与任何具有 XPRO 接口的单片机开发板搭配使用。必须获取 ATECC608A-TNGTLS 的特定示例才能与这些工具包搭配使用。

6.2.3 CryptoAuthLib

CryptoAuthLib 是一个软件库，支持 Microchip 的 CryptoAuthentication 器件系列。Microchip 建议在基于 ATECC608A-TNGTLS 进行开发时使用该库。该库实现了执行本数据手册中详述的命令所需的 API 调用。

该库已能够轻松与许多 Microchip 单片机搭配使用，而且还可通过硬件抽象层（Hardware Abstraction Layer，HAL）轻松扩展至其他单片机，包括其他供应商制造的单片机。

有关这些工具的更多信息，请查看以下相关信息：

- CryptoAuthLib——网络链接
- CryptoAuthLib——GitHub

API 调用
数据手册中的每条命令都有一个或多个与其关联的 API 调用。通常，命令都有一个基本的 API 调用，在其中可以指定所有输入参数。命令及其相关小节中显示的参数可与该命令搭配使用。每个 API 调用也都有不同的模式。下表列出了命令和基本 API 调用的示例。有关最准确的 API 信息，请参见 GitHub 信息。

表 6-1. CryptoAuthLib API 调用命令示例

器件命令	API 调用	注释
Info	atcab_info_base()	
Write	atcab_write()	
Read	atcab_read_zone()	
SHA	atcab_sha_base()	
Sign	atcab_sign_base()	
Random	atcab_random()	
Verify	atcab_verify()	

6.3 TrustFLEX 与 Trust&GO

Trust&GO 产品旨在让小规模生产的客户能够使用现成安全解决方案。该产品的简单载入过程利用了 Microchip 的安全制造解决方案和基础架构。通过使用该流程，客户不必自行创建安全制造环境。

但是，有时客户可能希望对他们的安全环境有更多控制权，同时仍希望享有 Trust&GO 产品的易用性。因此，我们为这类客户量身打造了 TrustFLEX 产品。TrustFLEX 产品不但能够实现 Trust&GO 产品的用例，而且为某些安全密钥和证书提供了额外的灵活性。

- 与 Trust&GO 产品的锁定配置相同。
- 与 Trust&GO 的数据槽定义相同。
- 能够将客户的公钥配置到器件中以实现安全引导。

- 能够进行对称密钥身份验证。客户可以安全地提供其所需的对称密钥，作为安全配置过程的一部分。
- 能够自定义证书元素并将证书链链接到客户所需的 PKI。
- 提供用于 I^2C 或 SWI 接口器件的选项。

有关 TrustFLEX 产品和其他配置选项的更多信息，请参见 Microchip CryptoAuthentication 网页。

7. I²C 接口

I²C 接口使用 SDA 和 SCL 引脚来指示 ATECC608A-TNGTLS 的各种 I/O 状态。该接口设计为在协议级别与工作在 1 MHz 下的 Microchip AT24C16 串行 EEPROM 兼容。

注： 两个器件之间有许多差别（例如，ATECC608A-TNGTLS 和 AT24C16 有不同的默认 I²C 地址）；因此，设计人员应仔细阅读相应的数据手册。

由于 ATECC608A-TNGTLS 的输出引脚上仅包含一个开漏驱动器，因此 SDA 引脚通常使用外部上拉电阻拉高。总线主器件可能是开漏型或图腾柱型。在后一种情况下，当 ATECC608A-TNGTLS 在总线上驱动结果时应当为三态。SCL 引脚为输入，必须始终由外部器件或电阻驱动为高电平和低电平。

7.1 I/O 条件

器件响应以下 I/O 条件：

7.1.1 器件休眠

当器件休眠时，它将忽略除唤醒状态以外的所有状态。

- 唤醒——在 SDA 保持低电平的时间 ≥ t_{WLO} 后，该器件将在 SDA 的上升沿退出低功耗模式。在 t_{WHI} 的延时后，它将准备好接收 I²C 命令。
- 当器件空闲或休眠时，在 t_{WLO} 期间，器件将忽略 SCL 引脚上的任何电平或转换。在 t_{WHI} 期间的某个时刻，将使能 SCL 引脚，并且将遵循 7.1.2 器件唤醒中列出的条件。

唤醒条件要求系统处理器手动将 SDA 引脚驱动为低电平并持续 t_{WLO}，或者以足够低的时钟速率传输数据字节 0x00 以使 SDA 的低电平时间持续最短周期 t_{WLO}。当器件唤醒时，正常的处理器 I²C 硬件和/或软件可用于器件通信。这包括使器件回到低功耗（即休眠）模式所需的 I/O 序列。

提示： 有一种方法可以轻松生成唤醒脉冲，即以 100 kHz 的频率发送字节 0x00。后续命令可以更高的频率运行。

在 I²C 模式下，器件将忽略器件已经唤醒时发送的唤醒序列。

总线上有多个器件

当总线上有多个器件且 I²C 接口的运行速度低于 300 kHz 左右时 [1]，传输某些数据模式将导致总线上的 ATECC608A-TNGTLS 器件唤醒。频率越低，器件唤醒的可能性就越高。由于沿总线传输的后续器件地址只匹配所需的器件，因此 ATECC608A-TNGTLS 不会响应但会唤醒。建议在以较低频率与另一器件通信后，发出休眠或空闲序列以使 ATECC608A-TNGTLS 回到已知状态。

重要： t_{WLO} 是指系统为确保 ATECC608A-TNGTLS 在所有制造和环境条件下唤醒而必须提供的最短时间。实际上，器件可由更小的脉冲宽度唤醒。

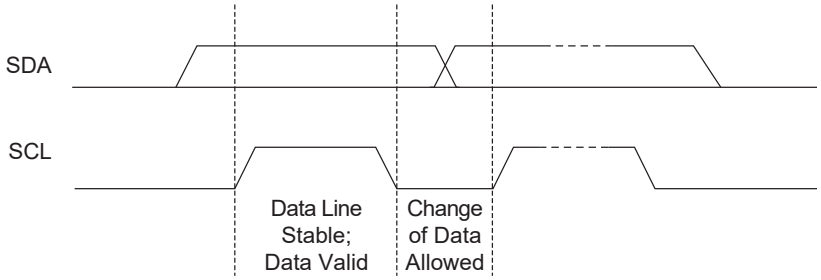
7.1.2 器件唤醒

当器件唤醒时，它将遵循下列条件：

- **数据 0**：如果 SDA 为低电平且保持稳定，而 SCL 由低电平变为高电平再变为低电平，则将在总线上传输一个 0 位。当 SCL 为低电平时，SDA 可发生变化。
- **数据 1**：如果 SDA 为高电平且保持稳定，而 SCL 由低电平变为高电平再变为低电平，则将在总线上传输一个 1 位。当 SCL 为低电平时，SDA 可发生变化。

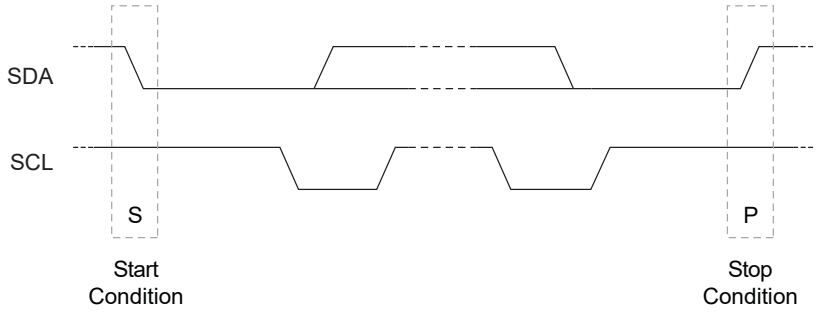
[1] 给定器件的实际频率将随过程和环境因素而变化。在所有条件下，该值都被认为是安全值。

图 7-1. I²C 接口上的数据位传输



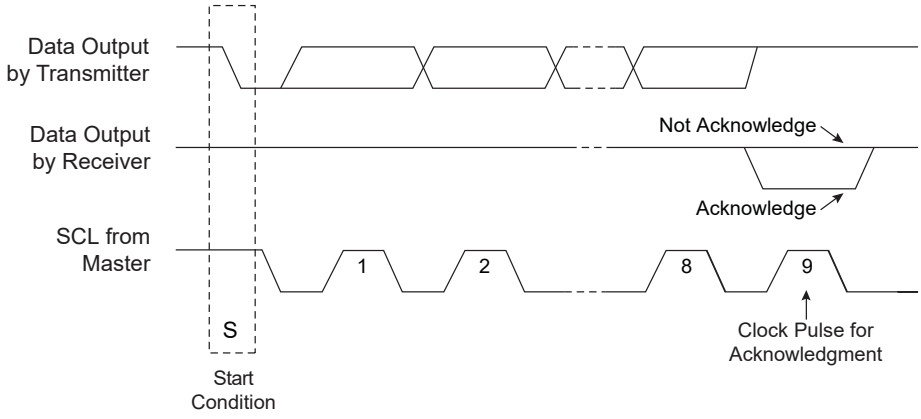
- **启动条件**：必须将 SDA 从高电平转换为低电平且 SCL 为高电平作为优先于所有命令的启动条件。
- **停止条件**：SDA 线从低电平转换为高电平且 SCL 为高电平为停止条件。器件收到此条件后，当前的 I/O 事务结束。在输入端，如果器件有足够的字节来执行命令，则器件转换到繁忙状态并开始执行。停止条件应始终在发送到器件的任何数据包结束时发送。

图 7-2. I²C 接口上的启动和停止条件



- **确认（ACK）**：在每个地址或数据字节传输后的第 9 个时钟周期，接收器将拉低 SDA 引脚以确认正确接收字节。
- **不确认（NACK）**：在每个地址或数据字节传输后的第 9 个时钟周期，接收器也可使 SDA 引脚保持高电平，以指示接收字节时出现问题，或者此字节完成组传输。

图 7-3. I²C 接口上的 NACK 和 ACK 条件



如果配置区域中的 I2C_Address 字节针对总线上的每个器件以不同方式编程，则多个 ATECC608A-TNGTLS 器件可轻松共用相同的 I²C 接口信号。由于器件地址的全部 7 位均可编程，ATECC608A-TNGTLS 还可将 I²C 接口与任何 I²C 器件（包括任何串行 EEPROM）共用。

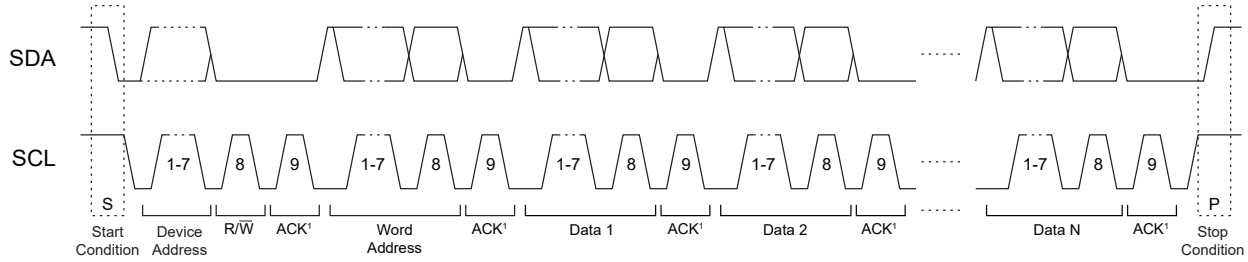
7.2 到 ATECC608A-TNGTLS 的 I²C 传输

下表总结了从系统到 ATECC608A-TNGTLS 的数据传输。传输顺序如下：

- 启动条件

- 器件地址字节
- 字地址字节
- 可选数据字节（1 至 N）
- 停止条件

图 7-4. 到 ATECC608A-TNGTLS 的正常 I²C 传输



ATECC608A-TNGTLS ACK 周期将 SDA 驱动为低电平。

下表标记了 I/O 事务的字节。标有“I²C 名称”的列提供了 AT24C16 数据手册中所描述字节的名称。**表 7-1. 到 ATECC608A-TNGTLS 的 I²C 传输**

名称	I²C 名称	说明
器件地址	器件地址	此字节选择 I²C 接口上的特定器件。如果此字节的 bit 1 至 bit 7 与配置区域中的 I2C_Address 字节的 bit 1 至 bit 7 匹配，则选择 ATECC608A-TNGTLS。此字节的 bit 0 是标准 I²C R/W 位，并且应为 0 以指示写操作（器件地址后的字节从主器件传输到从器件）。
字地址	字地址	要正常工作，此字节的值应为 0x03。有关更多信息，请参见 7.2.1 字地址值。
命令	数据 1,N	由计数、命令数据包和两个字节的 CRC 组成的命令组。CRC 通过大小和数据包字节计算。请参见 4.1 I/O 事务。

由于器件将命令输入缓冲区视为 FIFO，因此输入组可以通过一个或多个 I²C 命令组中发送到器件。发送到器件的第一个字节是计数，所以在器件接收到相应数量的字节之后，它将忽略随后接收的任何字节，直到执行完成。

系统必须在最后一个命令字节后发送一个停止条件，以确保 ATECC608A-TNGTLS 将启动命令的计算。未能发送停止条件可能最终导致同步丢失；有关恢复程序，请参见 7.2.2 I2C 同步。

7.2.1 字地址值

在 I²C 写数据包期间，ATECC608A-TNGTLS 会将发送的第二个字节解释为字地址，表示数据包功能，如下表所述：

表 7-2. 字地址值

名称	值	说明
复位	0x00	复位地址计数器。下一个 I²C 读取或写入事务将从 I/O 缓冲区起始处开始。
休眠（低功耗）	0x01	ATECC608A-TNGTLS 进入低功耗休眠模式，忽略所有后续的 I/O 转换，直到下一个唤醒标志。器件的整个易失性状态将复位。
空闲	0x02	ATECC608A-TNGTLS 进入空闲模式，忽略所有后续的 I/O 转换，直到下一个唤醒标志。TempKey、MessageDigestBuffer 和备用密钥寄存器的内容将保留。
命令	0x03	将后续字节写入输入命令缓冲区中先前写入内容之后的连续地址。这是正常操作。
保留	0x04-0xFF	这些地址不应发送到器件。

7.2.2 I²C 同步

系统可能会由于系统复位、I/O 噪声或其他条件失去与 ATECC608A-TNGTLS I/O 端口的同步。在这种情况下，ATECC608A-TNGTLS 可能不会按预期响应，可能处于休眠状态，也可能在系统期望发送数据的时间间隔期间传输数据。要重新同步，可按照以下步骤操作：

1. 为了确保 I/O 通道复位，系统必须发送标准 I²C 软件复位序列，具体如下：
   - 一个启动位条件。
   - 9 个 SCL 周期，通过系统上拉电阻使 SDA 保持高电平。
   - 另一个启动位条件。
   - 一个停止位条件。

   然后应当可以发送一个读序列，如果同步正确完成，ATECC608A-TNGTLS 将确认器件地址。在数据周期内，器件可能返回数据，也可能使总线悬空（系统会将其解释为数据值 0xFF）。

   如果器件确认了器件地址，系统应复位内部地址计数器，强制 ATECC608A-TNGTLS 忽略可能已发送的任何部分输入命令。这可以通过将一个写序列发送到字地址 0x00（复位），然后再发送一个停止条件来实现。
2. 如果器件不通过 ACK 来响应器件地址，则它可能处于休眠状态。在这种情况下，系统必须发送一个完整的 Wake 令牌并在上升沿后等待 t_{WHI}。系统随后可以发送另一个读序列，如果同步完成，器件将确认器件地址。
3. 如果器件仍不通过 ACK 来响应器件地址，则它可能忙于执行命令。系统必须等待最长 t_{EXEC}（最大值），然后发送读取序列，这将由器件确认。

7.3 休眠序列

系统完成使用 ATECC608A-TNGTLS 后，系统应发出休眠序列，使器件进入低功耗模式。此序列包含正确的器件地址，接着是值 0x01（作为字地址），然后是停止条件。这种到低功耗状态的转换会导致器件的内部命令引擎和输入/输出缓冲区完全复位。当器件唤醒且不忙时，此序列可随时发送到器件。

7.4 空闲序列

如果所需命令的总序列超过 $t_{WATCHDOG}$，则器件将自动进入休眠状态，并丢失存储在易失性寄存器中的任何信息。在看门狗时间间隔完成之前将器件置于空闲模式可防止此操作。当器件收到 Wake 令牌时，它将重新启动看门狗定时器，并继续执行。

此空闲序列包含正确的器件地址，接着是值 0x02（作为字地址），然后是停止条件。当器件唤醒且不忙时，此序列可随时发送到器件。

7.5 来自 ATECC608A-TNGTLS 的 I²C 传输

当 ATECC608A-TNGTLS 唤醒且不繁忙时，总线主器件可以使用 I²C 读取从器件中获取当前输出缓冲区内容。如果有效的命令结果可用，则返回的组大小由已经运行的特定命令决定。否则，组大小（以及返回的第一个字节）将始终为 4：计数、状态/错误和 2 字节 CRC。总线时序如图 8-2 所示。

表 7-3. 来自 ATECC608A-TNGTLS 的 I²C 传输

名称	I²C 名称	方向	说明
器件地址	器件地址	至从器件	此字节选择 I2C 接口上的特定器件，如果此字节的 bit 1 至 bit 7 与配置区域中的 I2C_Address 字节的 bit 1 至 bit 7 匹配，则选择 ATECC608A-TNGTLS。此字节的 bit 0 是标准 I²C R/W 引脚，并且应为 1 以指示器件地址后的字节从从器件传输到主器件（读操作）。
数据	数据 1,N	至主器件	由计数、状态/错误字节或输出数据包（后跟两字节的 CRC）组成的输出组。请参见 4.1 I/O 事务。

主器件可以重复读取状态、错误或命令输出。每次 Read 命令沿 I²C 接口发送到 ATECC608A-TNGTLS 时，器件均会发送输出缓冲区中的下一个连续字节。有关器件如何处理地址计数器的详细信息，请参见后续章节。

如果 ATECC608A-TNGTLS 处于繁忙、空闲或休眠状态，它将不会确认读序列上的器件地址。如果部分命令已经发送到器件，并且读取序列[启动 + 器件地址（R/W == R）]发送到器件，则 ATECC608A-TNGTLS 将不确认器件地址以指示没有数据可供读取。

8. 电气特性

8.1 绝对最大值

工作温度	-40°C 至+85°C
存储温度	-65°C 至+150°C
最大工作电压	6.0V
直流输出电流	5.0 mA
任意引脚上的电压-0.5V 至（VCC + 0.5V）	-0.5V 至（VCC + 0.5V）
ESD 额定值：	
人体模型（HBM）ESD	>4kV
充电器件模型（CDM）ESD	>1kV

注： 如果器件运行条件超过上述“绝对最大值”，可能对器件造成永久性损坏。上述数值仅是工作条件最大值，不代表器件在上述极限值或超出极限值的情况下仍可正常工作。器件长时间在绝对最大值条件下工作，其可靠性可能受到影响。

8.2 可靠性

ATECC608A-TNGTLS 采用 Microchip 公司具有极高可靠性的 CMOS EEPROM 制造技术制作。

表 8-1. EEPROM 可靠性

参数	最小值	典型值	最大值	单位
+85°C 下的耐写入次数（每个字节）	400,000	—	—	写周期
数据保持时间（+55°C 时）	10	—	—	年
数据保持时间（+35°C 时）	30	50	—	年
耐读取次数	无限			读周期

8.3 交流参数：所有 I/O 接口

图 8-1. 交流时序图：所有接口

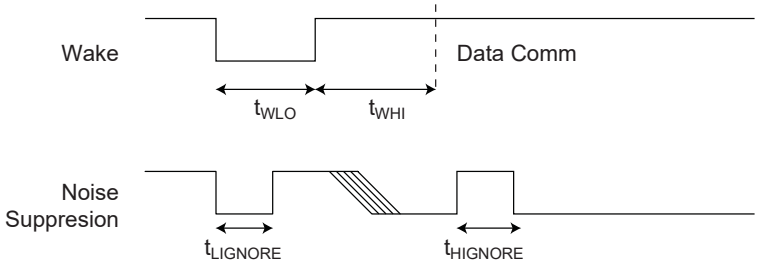


表 8-2. 交流参数：所有 I/O 接口

参数	符号	方向	最小值	典型值	最大值	单位	条件
上电延时(2)	t_{PU}	至加密器件	100	—	—	µs	从 V_{CC} > V_{CC} min 到 t_{WLO} 开始的最短时间。
唤醒为低电平的持续时间	t_{WLO}	至加密器件	60	—	—	µs	
唤醒为高电平到数据通信的延时	t_{WHI}	至加密器件	1500	—	—	µs	除非实现了轮询，否则 SDA 在整个过程中应保持稳定的高电平状态。上电时不使能 SelfTest。
使能 SelfTest 时的唤醒为高电平延时	t_{WHIST}	至加密器件	20	—	—	ms	除非实现了轮询，否则 SDA 在整个过程中应保持稳定的高电平状态。
上桥臂毛刺滤波器激活时间	$t_{HIGNORE_A}$	至加密器件	45(1)	—	—	ns	无论激活时的状态如何，宽度短于此时间的脉冲都将被器件忽略。
下桥臂毛刺滤波器激活时间	$t_{LIGNORE_A}$	至加密器件	45(1)	—	—	ns	无论激活时的状态如何，宽度短于此时间的脉冲都将被器件忽略。
下桥臂毛刺滤波器休眠时间	$t_{LIGNORE_S}$	至加密器件	15(1)	—	—	µs	处于休眠模式时，宽度短于此时间的脉冲将被器件忽略。
看门狗超时	$t_{WATCHDOG}$	至加密器件	0.7	1.3	1.7	s	从唤醒到强制器件进入休眠模式的时间（Config.ChipMode[2]为 0 时）。

注：

1. 这些参数为特性值，但未经生产测试。
2. 如果在配置区中使能了上电自检，则上电延时将显著延长。

8.3.1 交流参数：I²C 接口

图 8-2. I²C 同步数据时序

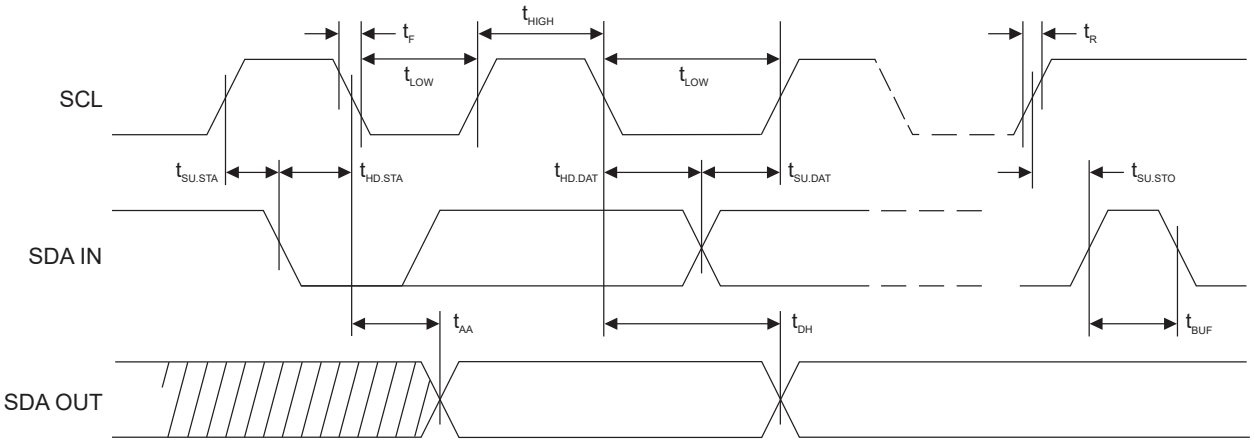


表 8-3. I2C 接口的交流特性

除非另外说明，否则适用的推荐工作范围为：T_A = -40°C 至+85°C，V_{CC} = +2.0V 至+5.5V，C_L = 1 TTL 栅极和 100 pF

参数	符号	最小值	最大值	单位
SCL 时钟频率	f_{SCL}	0	1	MHz
SCL 高电平时间	t_{HIGH}	400	—	ns

.....（续）

参数	符号	最小值	最大值	单位
SCL 低电平时间	t_{LOW}	400	—	ns
启动条件建立时间	$t_{SU.STA}$	250	—	ns
启动条件保持时间	$t_{HD.STA}$	250	—	ns
停止条件建立时间	$t_{SU.STO}$	250	—	ns
数据输入建立时间	$t_{SU.DAT}$	100	—	ns
数据输入保持时间	$t_{HD.DAT}$	0	—	ns
输入上升时间(1)	t_R	—	300	ns
输入下降时间(1)	t_F	—	100	ns
时钟低电平到数据输出有效的时间	t_{AA}	50	550	ns
数据输出保持时间	t_{DH}	50	—	ns
SMBus 超时延时	$t_{TIMEOUT}$	25	75	ms
在新传输开始前时间总线必须保持空闲的时间(1)	t_{BUF}	500	—	ns

注：

1. 上述值均为特性值，未经测试。
2. 交流测量条件：
   - R_L（连接 SDA 和 V_{CC}）：1.2 kΩ（对于 V_{CC} = +2.0V 至+5.0V）
   - 输入脉冲电压：$0.3V_{CC}$ 至 $0.7V_{CC}$
   - 输入上升和下降时间：≤ 50 ns
   - 输入和输出时序参考电压：$0.5V_{CC}$

8.4 直流参数：所有 I/O 接口

表 8-4. 所有 I/O 接口上的直流参数

参数	符号	最小值	典型值	最大值	单位	条件
环境工作温度	T_A	-40	—	+85	°C	
电源电压	V_{CC}	2.0	—	5.5	V	
电源工作电流	I_{CC}	—	2	3	mA	在 I/O 传输期间或执行非 ECC 命令期间等待 I/O。独立于时钟分频值。
		—	—	14	mA	在 ECC 命令执行期间。时钟分频值 = 0x0
电源空闲电流	I_{IDLE}	—	800	—	µA	当器件处于空闲模式时，V_{SDA} 和 V_{SCL} < 0.4V 或> V_{CC} – 0.4
休眠电流	I_{SLEEP}	—	30	150	nA	当器件处于休眠模式时，$V_{CC} \le 3.6V$， V_{SDA} 和 V_{SCL} < 0.4V 或> V_{CC} – 0.4，$T_A \le +55°C$
		—	—	2	µA	当器件处于休眠模式时。 在整个 V_{CC} 和温度范围内。
输出低电压	V_{OL}	—	—	0.4	V	当器件处于工作模式时，V_{CC} = 2.5 至 5.5V
输出低电流	I_{OL}	—	—	4	mA	当器件处于工作模式时，V_{CC} = 2.5 至 5.5V，V_{OL} = 0.4V

.....（续）

参数	符号	最小值	典型值	最大值	单位	条件
Theta JA	Θ_{JA}	—	166	—	°C/W	SOIC（SSH）
		—	173	—	°C/W	UDFN（MAH）
		—	146	—	°C/W	RBH

8.5 V_{IH} 和 V_{IL} 规范

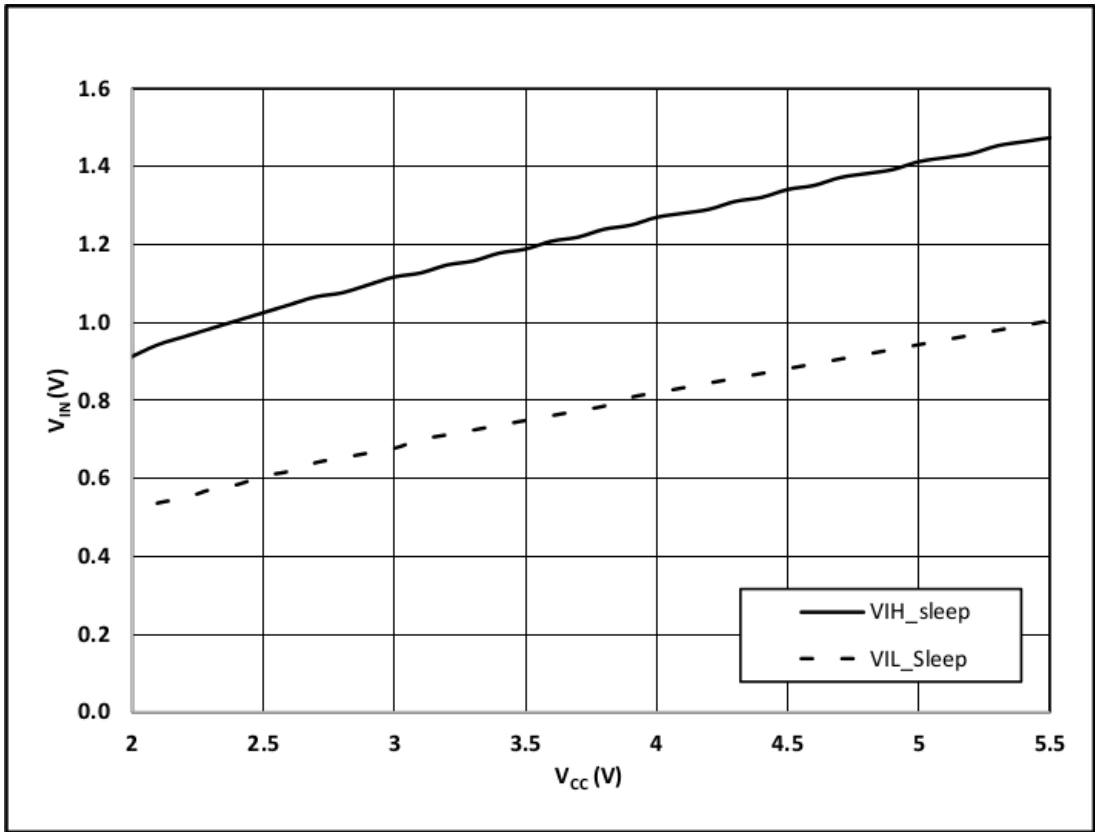
器件的输入电平将根据器件的模式和电压而变化。休眠或空闲模式下的输入电压阈值取决于 V_{CC} 电平，如图 8-3 所示。在休眠或空闲模式下时，TTLenable 位不起作用。

ATECC608A-TNGTLS 的有效输入电平固定，不会随 V_{CC} 电平变化。发送到器件的输入电平必须符合下表。

表 8-5. 所有 I/O 接口上的 V_{IL} 和 V_{IH}（TTLenable = 0 时）

参数	符号	最小值	典型值	最大值	单位	条件
输入低电压	V_{IL}	-0.5	—	0.5	V	当器件处于工作状态且配置存储器中的 TTLenable 位为 0 时；否则请参见上文。
输入高电压	V_{IH}	1.5	—	V_{CC} + 0.5	V	当器件处于工作状态且配置存储器中的 TTLenable 位为 0 时；否则请参见上文。

图 8-3. 休眠和空闲模式下的 V_{IH} 和 V_{IL}



9. 封装图

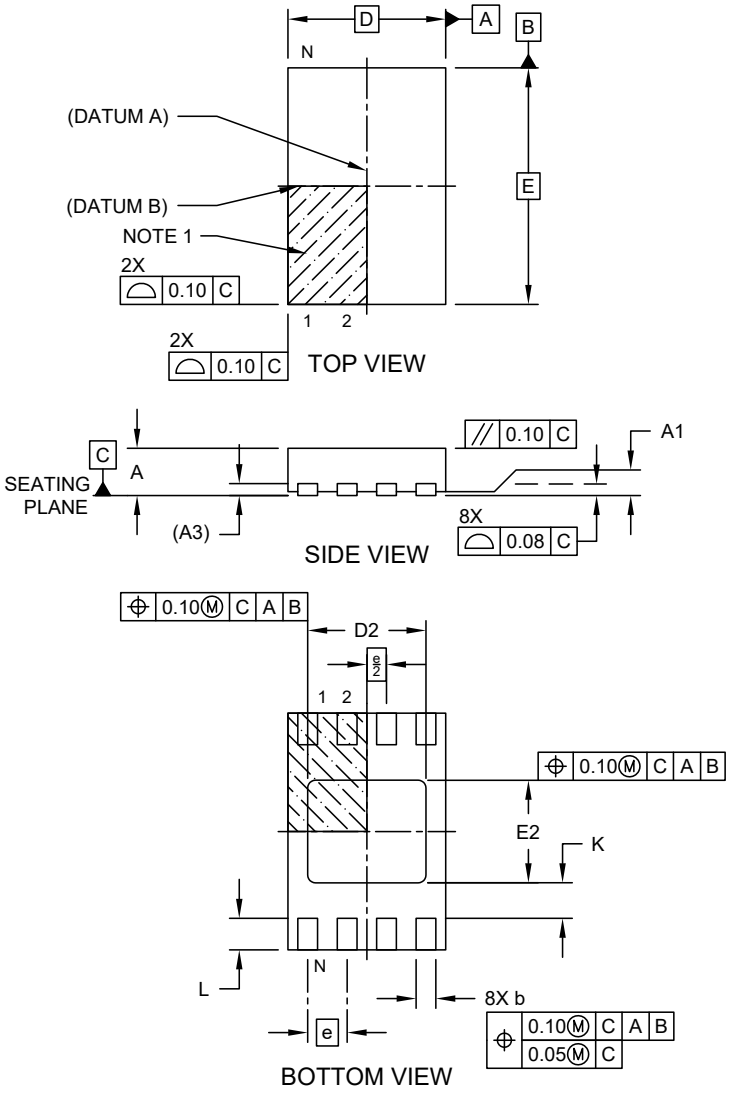
9.1 封装标识信息

为保证 Microchip 的整体安全性，所有加密器件的部件标识有意进行了模糊处理。封装顶部的标识不会提供有关实际器件类型或器件制造商的任何信息。封装上的字母数字代码可提供制造信息，并因装配批次而异。不应在任何来料检验过程中使用此封装标识。

9.2 8 焊盘 UDFN

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy YNZ Package

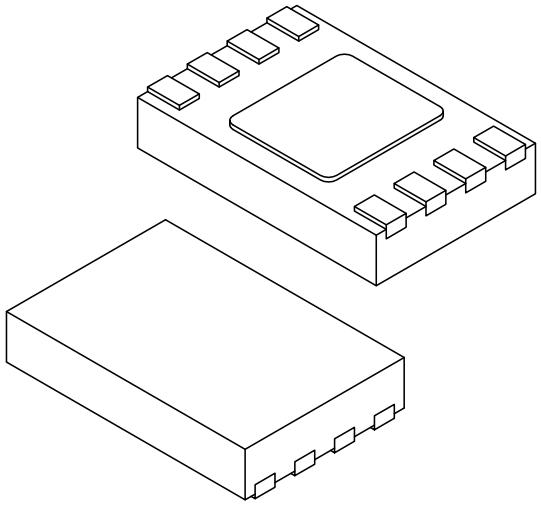
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-21355-Q4B Rev A Sheet 1 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy YNZ Package

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		8	
Pitch	e		0.50 BSC	
Overall Height	A	0.50	0.55	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.152 REF	
Overall Length	D		2.00 BSC	
Exposed Pad Length	D2	1.40	1.50	1.60
Overall Width	E		3.00 BSC	
Exposed Pad Width	E2	1.20	1.30	1.40
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

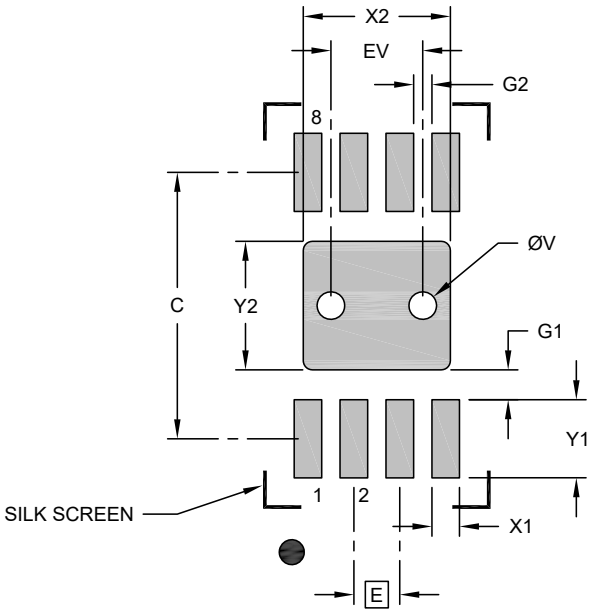
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21355-Q4B Rev A Sheet 2 of 2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (Q4B) - 2x3 mm Body [UDFN]
Atmel Legacy YNZ Package

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	X2			1.60
Optional Center Pad Length	Y2			1.40
Contact Pad Spacing	C		2.90	
Contact Pad Width (X8)	X1			0.30
Contact Pad Length (X8)	Y1			0.85
Contact Pad to Center Pad (X8)	G1	0.20		
Contact Pad to Contact Pad (X6)	G2	0.33		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

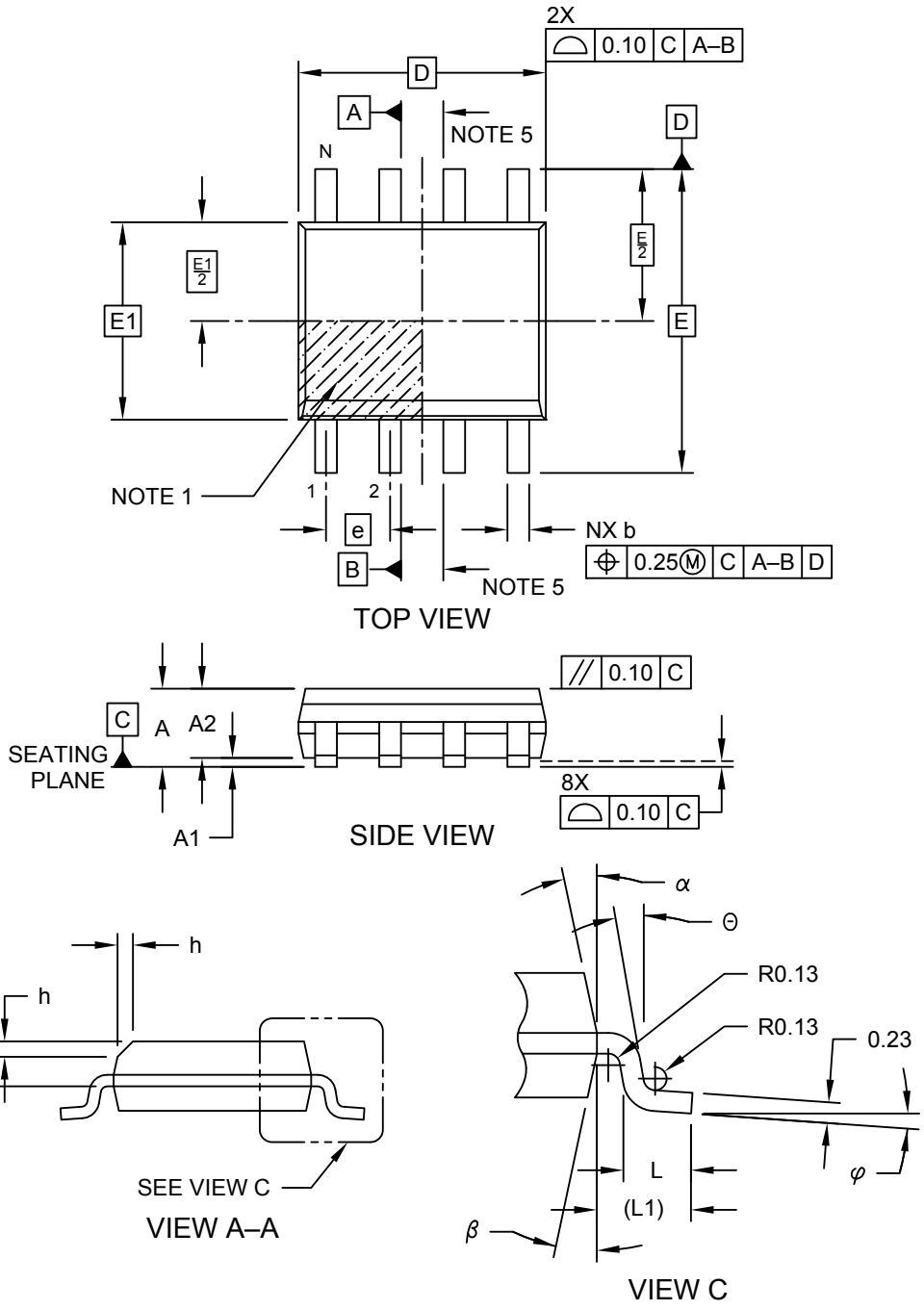
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-21355-Q4B Rev A

9.3 8 引脚 SOIC

8-Lead Plastic Small Outline - Narrow, 3.90 mm (.150 In.) Body [SOIC]
Atmel Legacy Global Package Code SWB

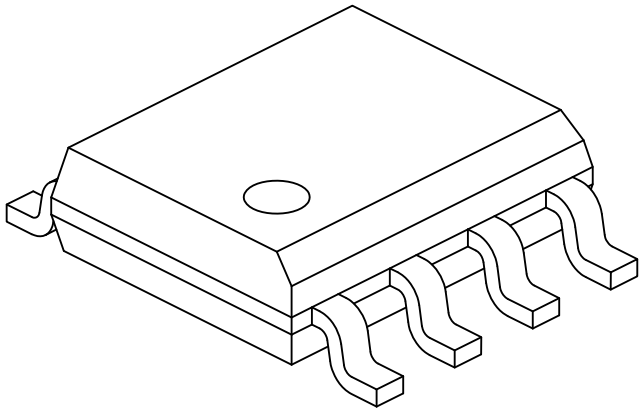
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing No. C04-057-SWB Rev E Sheet 1 of 2

8-Lead Plastic Small Outline - Narrow, 3.90 mm (.150 In.) Body [SOIC]
Atmel Legacy Global Package Code SWB

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		8	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		4.90 BSC	
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1		1.04 REF	
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.17	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

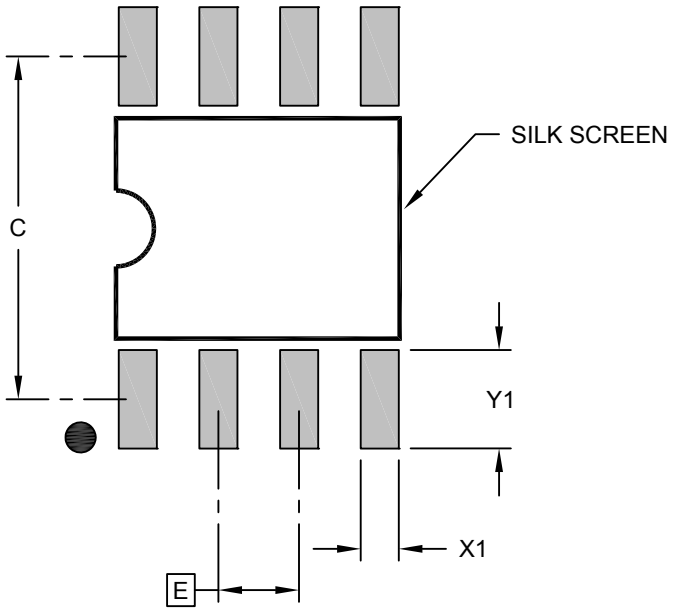
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-057-SWB Rev E Sheet 2 of 2

8-Lead Plastic Small Outline - Narrow, 3.90 mm (.150 In.) Body [SOIC]
Atmel Legacy Global Package Code SWB

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width (X8)	X1			0.60
Contact Pad Length (X8)	Y1			1.55

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2057-SWB Rev E

10. 版本历史

版本 A（2019 年 8 月）
本文档的初始版本。

版本 B（2019 年 9 月）
将配置信息更正为适当的 ATECC608A-TNGTLS 配置。

- Slot 5、7、13-15 设置为保留

Microchip 网站

Microchip 网站（http://www.microchip.com/）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答（FAQ）、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请访问 http://www.microchip.com/pcn，然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（ESE）
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 http://www.microchip.com/support 获得网上技术支持。

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

PART NO. X -X
Device Package Type Tape and Reel

器件：	ATECC608A-TNGTLS：具有基于硬件的安全密钥存储功能的预配置密码协处理器，面向物联网市场。	
封装选项	U	8 焊盘（主体 2 x 3 x 0.6 mm）增强散热型塑封超薄双列扁平无引线封装（UDFN）
	S	8 引脚（主体宽 0.150”）塑封鸥翼小外形封装（JEDEC SOIC）
卷带式选项	G	2k 卷盘
	C	100 件卷盘
	B	10 件装——原型器件

示例：

- ATECC608A-TNGTLSU-G：Trust&GO TLS，已配置，8-UDFN，I^2C，2K 卷盘
- ATECC608A-TNGTLSU-C：Trust&GO TLS，已配置，8-UDFN，I^2C，100 件卷盘
- ATECC608A-TNGTLSU-B：Trust&GO TLS，原型设计，8-UDFN，I^2C，10 件装
- ATECC608A-TNGTLSS-G：Trust&GO TLS，已配置，8-SOIC，I^2C，2K 卷盘
- ATECC608A-TNGTLSS-C：Trust&GO TLS，已配置，8-SOIC，I^2C，100 件卷盘
- ATECC608A-TNGTLSS-B：Trust&GO TLS，原型设计，8-SOIC，I^2C，10 件装

注：

1. 卷带式标识符仅出现在产品目录的部件编号描述中。该标识符用于订货目的，不会印刷在器件封装上。关于包装是否提供卷带式选项的信息，请咨询当地的 Microchip 销售办事处。

Microchip 器件代码保护功能

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc.及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc.的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，否则在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PackeTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total

Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 均为 Microchip Technology Inc.在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc.的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2020，Microchip Technology Incorporated 版权所有。

ISBN：978-1-5224-5723-7

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 http://www.microchip.com/quality。

全球销售及服务网点

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话：480-792-7200 传真：480-792-7277 技术支持： http://www.microchip.com/support 网址： http://www.microchip.com **亚特兰大** 德卢斯，佐治亚州 电话：678-957-9614 传真：678-957-1455 **奥斯汀，德克萨斯州** 电话：512-257-3370 **波士顿** 韦斯特伯鲁，马萨诸塞州 电话：774-760-0087 传真：774-760-0088 **芝加哥** 艾塔斯卡，伊利诺伊州 电话：630-285-0071 传真：630-285-0075 **达拉斯** 阿迪森，德克萨斯州 电话：972-818-7423 传真：972-818-2924 **底特律** 诺维，密歇根州 电话：248-848-4000 **休斯顿，德克萨斯州** 电话：281-894-5983 **印第安纳波利斯** 诺布尔斯维尔，印第安纳州 电话：317-773-8323 传真：317-773-5453 电话：317-536-2380 **洛杉矶** 米慎维荷，加利福尼亚州 电话：949-462-9523 传真：949-462-9608 电话：951-273-7800 **罗利，北卡罗来纳州** 电话：919-844-7510 **纽约，纽约州** 电话：631-435-6000 **圣何塞，加利福尼亚州** 电话：408-735-9110 电话：408-436-4270 **加拿大 - 多伦多** 电话：905-695-1980 传真：905-695-2078	**澳大利亚 - 悉尼** 电话：61-2-9868-6733 **中国 - 北京** 电话：86-10-8569-7000 **中国 - 成都** 电话：86-28-8665-5511 **中国 - 重庆** 电话：86-23-8980-9588 **中国 - 东莞** 电话：86-769-8702-9880 **中国 - 广州** 电话：86-20-8755-8029 **中国 - 杭州** 电话：86-571-8792-8115 **中国 - 香港特别行政区** 电话：852-2943-5100 **中国 - 南京** 电话：86-25-8473-2460 **中国 - 青岛** 电话：86-532-8502-7355 **中国 - 上海** 电话：86-21-3326-8000 **中国 - 沈阳** 电话：86-24-2334-2829 **中国 - 深圳** 电话：86-755-8864-2200 **中国 - 苏州** 电话：86-186-6233-1526 **中国 - 武汉** 电话：86-27-5980-5300 **中国 - 西安** 电话：86-29-8833-7252 **中国 - 厦门** 电话：86-592-238-8138 **中国 - 珠海** 电话：86-756-321-0040	**印度 - 班加罗尔** 电话：91-80-3090-4444 **印度 - 新德里** 电话：91-11-4160-8631 **印度 - 浦那** 电话：91-20-4121-0141 **日本 - 大阪** 电话：81-6-6152-7160 **日本 - 东京** 电话：81-3-6880-3770 **韩国 - 大邱** 电话：82-53-744-4301 **韩国 - 首尔** 电话：82-2-554-7200 **马来西亚 - 吉隆坡** 电话：60-3-7651-7906 **马来西亚 - 槟榔屿** 电话：60-4-227-8870 **菲律宾 - 马尼拉** 电话：63-2-634-9065 **新加坡** 电话：65-6334-8870 **台湾地区 - 新竹** 电话：886-3-577-8366 **台湾地区 - 高雄** 电话：886-7-213-7830 **台湾地区 - 台北** 电话：886-2-2508-8600 **泰国 - 曼谷** 电话：66-2-694-1351 **越南 - 胡志明市** 电话：84-28-5448-2100	**奥地利 - 韦尔斯** 电话：43-7242-2244-39 传真：43-7242-2244-393 **丹麦 - 哥本哈根** 电话：45-4485-5910 传真：45-4485-2829 **芬兰 - 埃斯波** 电话：358-9-4520-820 **法国 - 巴黎** 电话：33-1-69-53-63-20 传真：33-1-69-30-90-79 **德国 - 加兴** 电话：49-8931-9700 **德国 - 哈恩** 电话：49-2129-3766400 **德国 - 海尔布隆** 电话：49-7131-72400 **德国 - 卡尔斯鲁厄** 电话：49-721-625370 **德国 - 慕尼黑** 电话：49-89-627-144-0 传真：49-89-627-144-44 **德国 - 罗森海姆** 电话：49-8031-354-560 **以色列 - 若那那市** 电话：972-9-744-7705 **意大利 - 米兰** 电话：39-0331-742611 传真：39-0331-466781 **意大利 - 帕多瓦** 电话：39-049-7625286 **荷兰 - 德卢内市** 电话：31-416-690399 传真：31-416-690340 **挪威 - 特隆赫姆** 电话：47-7288-4388 **波兰 - 华沙** 电话：48-22-3325737 **罗马尼亚 - 布加勒斯特** 电话：40-21-407-87-50 **西班牙 - 马德里** 电话：34-91-708-08-90 传真：34-91-708-08-91 **瑞典 - 哥德堡** 电话：46-31-704-60-40 **瑞典 - 斯德哥尔摩** 电话：46-8-5090-4654 **英国 - 沃金厄姆** 电话：44-118-921-5800 传真：44-118-921-5820

02/28/20