Keywords: MAXQ1061, MAXQ1062, security, TLS, IOT, TPM, embedded, cryptography, AES, ECDSA, digital signatures, authentication, secure download, random numbers, SP800-90A, secure element, secure storage, x509, secure provisioning, PKI, TCG

APPLICATION NOTE 6762

FUNDAMENTAL ADVANTAGES OF THE MAXQ1061/MAXQ1062 COMPARED TO CHIPS BASED ON THE TPM 2.0 STANDARD

By: Stéphane Di Vito and Christophe Tremlet

Abstract: Some security ICs are designed to be used as companion ICs of application processors. The Maxim® MAXQ1061/MAXQ1062 family and the Trusted Computing Group™-defined Trusted Platform Modules (TPM) are such companion chips. Security goals in IoT relate to device and server authentication, sensitive data protection, confidentiality and integrity of communications (e.g., TLS protocol), device integrity, and intellectual property protection.
In a more concrete example, an IoT node device needs to have a secure bootloader and secure firmware update, to send sensor data to a server over a TLS connection, and to store sensitive data in flash memory. The MAXQ1061/MAXQ1062 are specifically designed for small embedded systems, require small resources in the device's application processor, and are easy to use. This application note shows in many ways that using the MAXQ1061/MAXQ1062 is simpler and more efficient than using chips based on the TPM standard.

Introduction

Historically, chips based on the Trusted Computing Group™ (TCG)-defined Trusted Platform Module (TPM) standard are tamper-resistant, discrete, cryptographic co-processors that have been implemented into most consumer personal computers and servers. The security concepts in the chips based on the TPM standard have been designed many years ago to enable trust between computers exchanging data over a network, and to protect a user's data at rest against loss or theft. The chips based on the TPM standard are tamper-resistant, secure, cryptographic processors designed to perform cryptographic operations and securely store a small amount of data, including keys.

A dedicated hardware IP based on the TPM standard can also be implemented on an existing component or as firmware that leverages the trusted execution environment (e.g., Intel® TXT or Arm® TrustZone®) of an existing CPU. This application note is only considering discrete chips based on the TPM standard because of the proposed higher security level in terms of tamper resistance.

Chips based on the TPM standard are designed for a simple purpose, but they can also support the implementation of very complex security policies that define who can do what and when. This complexity, justified by the variety of scenarios required in computer applications, makes the learning curve of the TPM standard quite stiff. Verifying the security of the system is much more difficult due to this complexity. Even though chips based on the TPM standard are also marketed for embedded systems, their initial design and platform resource requirements make them difficult to integrate into simple systems as often seen in the IoT.

The MAXQ1061/MAXQ1062 are created by Maxim Integrated, based on requirements derived from internal, customer, and public, cryptographic-modules specifications, especially for simple, embedded, connected devices. The MAXQ1061/MAXQ1062 are designed to bring high-security certificate, key and data secure storage, and secure cryptographic primitives to low-cost embedded systems. The devices can be used for device integration into public key infrastructure f, device and server authentication, confidentiality and integrity of data, and device integrity. The MAXQ1061/MAXQ1062 provide a strict necessary set of 40 commands to achieve such purposes and to avoid unneeded complexity. The simple access control policy and feature set answer most security scenarios relevant to the IoT devices in a much simpler way than the chips based on the TPM standard.

The cryptographic algorithms provided by the MAXQ1061/MAXQ1062 conform to standards (e.g., NIST, ANSI, BSI). The MAXQ1061/MAXQ1062 are also tamper-resistant ICs using state-of-the-art anti-hacking countermeasures.

In this application note, we enumerate features and use cases related to relevant IoT devices and compare their implementation with the MAXQ1061/MAXQ1062 and the chips based on the TPM standard. Although, not all the infinite possibilities offered by a chip based on the TPM standard are going to be exposed.

Definitions

The following terms are used throughout this application note:

- Host: Main processor such as the CPU or SoC running the device
- Server: Remote computer to which the device connects
- Security IC: The MAXQ1061/MAXQ1062 or a chip based on the TPM standard

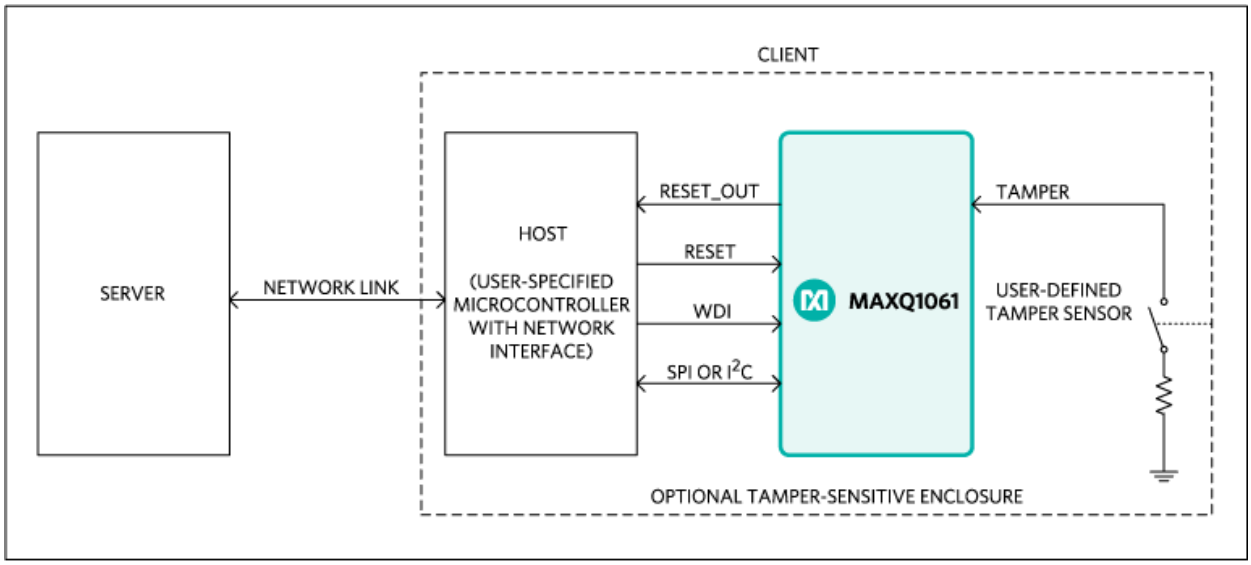

Figure 1. Example of integration of the MAXQ1061/MAXQ1062.

The MAXQ1061/MAXQ1062 are connected to a host processor using a SPI or I2C bus. Optionally, the host processor can control the RESET input and watchdog input (WDI) of the MAXQ1061/MAXQ1062. The MAXQ1061/MAXQ1062 can also be connected to a device-level tamper-detection mechanism (e.g., switches). Lastly, the MAXQ1061/MAXQ1062 can control the RESET input of the host processor.

Remember that security ICs behave as slaves. They receive commands from a host processor through a command bus, process the commands, and return the answer back. Those chips do not send anything spontaneously, nor do they actively read external memories, etc.

Purpose of Security ICs

The purpose of security ICs, or secure elements, is to increase the security of the key storage and to provide secure cryptographic implementations resistant to fault injection and side channel attacks. In addition, security ICs allow a strict isolation from other software running on the platform, since the only interaction channel is the command bus, which drastically reduces the attack surface and the exposure of the sensitive functions and data to the host processor's software. Besides, it is not easy to find implementations of off-the-shelf, pure software, cryptographic algorithms that are resistant to attacks (and even less as easy to develop them by yourself), unless the host processor already features integrated crypto-dedicated hardware blocks, and the silicon vendor provides such cryptographic routines.

A device can use a security IC to support data protection and security scenarios that software alone cannot achieve. For example, software cannot reliably report whether malware is present during the system startup process. The close integration between the security IC and the host processor makes the startup process clearer and easier to assess security wise. Evaluation of the device integrity is enabled through reliable measurement and reporting by the trusted software that starts the device.

Implementation of a security IC as part of a device provides a hardware root of trust, meaning it always behaves in a trusted way. For example, a private key stored in the security IC truly cannot leave the IC, or the certificates stored there cannot be tampered with.

Last, but not least, the security IC offloads the host processor from the memory footprint and computation resources required by the security functions, which is especially beneficial in devices featuring a small host processor such as the Arm Cortex®-M0 microcontrollers (typically 128K flash, 16K RAM, 40MHz).

Comparison of Main Characteristics

Package

The MAXQ1061/MAXQ1062 and the chips based on the TPM standard are supplied in similar packages. Package size is becoming critical in most embedded devices where PCB sizes become more reduced due to more constraining device form factors.

The MAXQ1061/MAXQ1062 come in a compact TSSOP14 6.4x4.9 package.

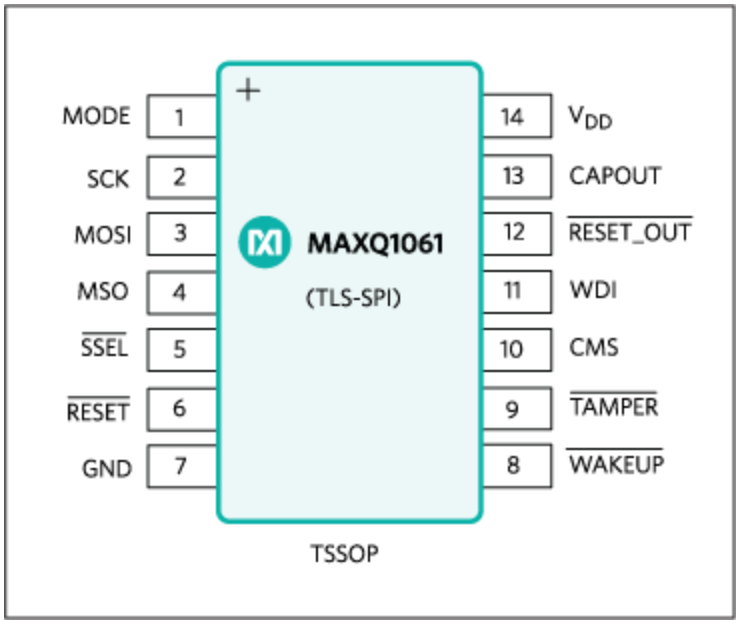

Figure 2. MAXQ1061/MAXQ1062 package.

Most chips based on the TPM standard come in TSSOP28 or VQFN 5x5 packages.

Power Supply and Consumption

Low power consumption in standby mode is a key advantage for battery powered devices. The MAXQ1061/MAXQ1062 have a typical standby current of 25µA, which is much lower than the considered chips based on the TPM standard that draw typically more than 100µA and often up to 300µA. The MAXQ1061/MAXQ1062 have a 30ms boot time and can be completely powered off when not used but still made available very quickly when needed.

In active mode, chips based on the TPM standard and the MAXQ1061/MAXQ1062 have a comparable active supply current of about 25mA. Supply voltage is 3.3V for the MAXQ1061/MAXQ1062 and most of the chips based on the TPM standard. Some chips based on the TPM standard offer 1.8V.

Internal Storage Size

The internal non-volatile storage of security ICs is essential to store sensitive data such as confidential data, secret or private keys, and data that must be preserved against modification such as certificates.

The MAXQ1061/MAXQ1062 have an internal nonvolatile tamper-resistant memory to store sensitive data. The MAXQ1061/MAXQ1062 have enough internal memory to serve most use cases applicable to embedded systems such as storing X.509 certificates, pairs of ECDSA keys, or other secret keys or arbitrary data. This internal storage is organized as a simple file system gated by a user defined access control. Please refer to the Designing Security Policies with Access Control section for more details.

The chips based on the TPM standard also have internal, tamper-proof, nonvolatile storage. This storage, among other things, allows to store root keys that encrypt other child keys, and child keys can, in turn, encrypt other grandchild keys. Chips based on the TPM standard can handle large key stores. The keys, other than the root keys, are usually stored outside the chip based on the TPM standard and are re-injected and decrypted into the chip whenever they are needed. The host side software handles this operation. Child keys are decrypted once loaded in the chip, then used, and eventually discarded, but the long-term key stays in the external encrypted storage. This feature is useful in a PC platform where numerous applications might want to keep a large number of keys for various purposes but most often relevant to IoT devices.

Communication Protocol

The MAXQ1061/MAXQ1062 provide an I^2C and a SPI interface that fit with most embedded designs. Chips based on the TPM standard also provide SPI interfaces and other computer-oriented interfaces such as LPC or SMBUS (close to I^2C). The LPC interface is dedicated to PC architectures and is barely applicable to embedded systems.

Host-Side Support Software

The TPM standard-compliant host-side software is large and complex, and the learning curve is stiff with very few programming resources available on the Internet. This software is dedicated to large systems featuring an operating system such as Linux® or Windows®. The TPM standard-compliant host-side software has the following major components:

- The TCG Software Stack (TSS). This software requires large systems such as Linux or Windows. Usually, the TPM standard-compliant software is hidden behind high-level tools such as a Microsoft® Crypto Provider or a PKCS#11 engine. While it handles the oddities and complexity of the low-level TPM standard-compliant communication, the specification of the TSS remains huge and complex.
- The TPM-standard-compliant driver. This driver is a lower-level component on which the TSS lies. The API of this driver is very complicated and sits at a very low level, making it even more difficult to implement in complex use cases. While the TPM standard integration in Windows and Linux is available out of the box, it is not easy, maybe even impossible, to port it on a bare metal platform.
- Feature API. On top of the existing layers, the TPM standard also proposes a Feature API, with a simpler, more integrated usage that limits to the most common scenario and provides about 45 API functions. However, the security policy definition is still complex.

Table 1 shows the results from our experimentation on a TSS (**TSS2**) on a Linux 64-bit target.

Table 1. TPM TSS Test Results

Text (bytes)	Static Data (bytes)*	Filename
146344	4972	./src/tss2-sys/.libs/libtss2-sys.so
253071	2140	./src/tss2-mu/.libs/libtss2-mu.so
16214	928	./src/tss2-tcti/.libs/libtss2-tcti-device.so
20019	960	./src/tss2-tcti/.libs/libtss2-tcti-mssim.so
600190	6640	./src/tss2-esys/.libs/libtss2-esys.so

* Dynamically allocated memory is not accounted here.

The code sizes in Table 1 do not fit with numerous small embedded devices that typically have less than 256KB of code memory and 64KB of RAM.

On the other hand, the MAXQ1061/MAXQ1062 need very simple software, which is suitable for the smallest microcontrollers. The host-side software is delivered in source, allowing for customization. It is a library in C language that offers functions that directly map to the command set of the MAXQ1061/MAXQ1062. Thanks to this reduced set of commands, complex use cases can be implemented in a rather straightforward way using simple programs. The odds of the communication protocol and bus level command formatting are hidden by the library, which exposes a clear, easy to use programmer's interface in C. This software relies on standard SPI or I^2C drivers of the platform. A minimalistic "glue" layer might have to be developed by the integrator to link the MAXQ1061/MAXQ1062 host library to the driver API of the microcontroller.

The MAXQ1061/MAXQ1062 host library typically uses around 6KB of code and 4KB of RAM and can be built for any small processor such as the Arm Cortex-M0. On top of this library, customized TLS client stacks are provided, such as the mbedTLS for any kind of microcontrollers (e.g., Cortex-M0, microcontrollers similar to high-end microcontrollers) and OpenSSL® for larger microcontrollers running embedded Linux or Windows. The TLS client stacks leverage the MAXQ1061/MAXQ1062 TLS capabilities.

Security IC Use Cases

This section lists use cases that the chips based on the TPM standard and MAXQ1061/MAXQ1062 can achieve.

Offloading Host-Processor Cryptographic Functions

Security ICs offload the main processor from sensitive cryptographic operations. The comparison of low-level cryptographic algorithms supported by each family of ICs is shown in **Table 2**.

Table 2. Supported Low-level Cryptographic Algorithms

Low-Level Cryptographic Features	Chips Based on the TPM Standard	MAXQ1061/MAXQ1062

Standard Cryptographic Algorithms	Yes	Yes
AES	AES-256 is optional in the TPM 2.0 standard	ECB, CBC, CCM 128/192/256 ECB/GCM 128, fast
Random Number Generator	SP800-90A, AIS31PTG2	Designed for SP800-90A SHA256DRBG
Key Generation	RSA, ECDSA	ECDSA
Secure Hash Algorithm	SHA-1, 256	SHA-1, 256, 384, 512
RSA Signature	2048-bit	No
RSA Encryption	2048-bit	No
ECDSA Signature	NIST® P-256	NIST P-256, 384, 521Brainpool 256, 384, 512
ECDH	Same curves as ECDSA	Same curves as ECDSA
HMAC	SHA-256	SHA-256, 384, 512
AES based MAC	No	Fast AES-CBC-MAC, AES-CMAC AES-GMAC with 128-bit keys on a dedicated SPI interface
TLS 1.2 PRF	No	SHA-256-based

* The design for this random number generator standard is not yet certified.

Fast AES Engine
The MAXQ1061/MAXQ1062 provide a fast AES engine over SPI for AES ECB and AES GCM encryption with 128-bit keys. The encryption speed goes up to 10MBps. This engine offloads the main processor from some symmetric encryption duties. Furthermore, it avoids exporting the AES keys to the main processor where the key could be exposed and revealed by attacks such as side channel attacks where the key is retrieved through power consumption or electromagnetic emission analysis. By using the onboard fast AES engine, the key is safely stored and manipulated in the MAXQ1061/MAXQ1062.

Chips based on the TPM standard do not feature such a high-speed symmetric-encryption engine, which is often useful when the host processor is not fast enough or is busy with other important tasks.

RSA vs ECDSA
Unlike the chips based on the TPM standard, the MAXQ1061/MAXQ1062 do not support RSA. However, RSA is getting more and more deprecated due to the long keys and digital signatures that can be several thousands of bits long compared to ECDSA, where keys and signatures are only a few hundred bits long.

TLS 1.2

The chips based on the TPM standard have no specific features to handle the TLS protocol. They can be used as basic cryptographic engines and long-term key storages, leveraged by certain TLS software stacks, such as wolfSSL®, to perform atomic operations like certificate generation, signature verification, AES encryption or decryption, HMAC signature or verification, or ECDH as needed by the TLS handshake and record processing. With such chips, TLS session keys are computed in the host processor, which is more exposed to hackers.

The MAXQ1061/MAXQ1062 can process a complete TLS 1.2 session from handshake to secure application data exchange while keeping the TLS session keys internal to the volatile memory. The TLS record layer is used when exchanging application data over TLS, and encryption and signature with negotiated keys is applied. This TLS record layer can be completely handled by the MAXQ1061/MAXQ1062, allowing the TLS session keys derived from the handshake phase to remain safe inside the chip and never exposed outside. The MAXQ1061/MAXQ1062, bundled with the provided TLS software stacks (i.e., mbedTLS, OpenSSL), are a turnkey solution that provide designers with secure communication for connected devices.

Table 3. TLS 1.2 Support

	Chips Based on the TPM 2.0 Standard	MAXQ1061/MAXQ1062
TLS 1.2 PRF	No	Yes
Protect session keys	No	Yes
Full handshake	No	Yes
TLS 1.2 record processing	No	Yes

External Memory Encryption

External memory encryption needs a symmetric key for encrypting/decrypting disk data. Symmetric (e.g., AES) is required for speed reasons. The security IC must hold the disk encryption/decryption key as encrypted and release it to the main CPU only if the CPU has booted into a secure state.

For chips based on the TPM standard, the secure state is determined using the PCR values (see the Secure Boot section for more information about platform configuration registers) that get updated during the bootstrap process. Then the TPM-standard chip releases the key to the host processor, which performs the disk encryption/decryption by itself because the chip is too slow for such an operation. When transferred to the host processor, the key might be exposed and no longer benefit from the protection of the chips based on the TPM standard. Side channel attacks (see the TLS 1.2 section) on the host processor or software attacks could reveal the key, and the whole system could be compromised.

The MAXQ1061/MAXQ1062 implement this behavior by simply using its secure boot feature and an internal memory object that holds the disk decryption key. When the platform has been securely booted, the MAXQ1061/MAXQ1062 can choose to transfer the key to the external AES-SPI engine and perform

high-speed encryption/decryption of the external memory instead of transferring the key to the host processor. This feature allows high speed decryption of external memory without exposing the decryption key. The option to transfer the key to the host processor is also available.

Device Identification and Authentication, Especially for Device Anti-Cloning

Public-key based identification and authentication relies on the presence of a truly unique identifier and a unique pair of keys in each device.

The MAXQ1061/MAXQ1062 can store unique pairs of ECDSA keys, which consist of one private and one public key in the internal nonvolatile memory. A certificate for the public key is also stored for device authentication. Upon request, the MAXQ1061/MAXQ1062 can use the private key to sign a challenge received from a peer and prove its identity, which is stored in the certificate, by sending back the signature of this challenge and the certificate. The authenticating peer can verify the certificate coming from the MAXQ1061/MAXQ1062 using a certification authority (CA) certificate, and then use the MAXQ1061/MAXQ1062 public key from that certificate to verify the challenge's signature. Upon success, the device that contains the MAXQ1061/MAXQ1062 is identified and authenticated.

Certificate Pinning

Storing certification authority (CA) certificates in a software vault on a device's standard microcontroller presents a risk. If an attacker can modify the software and get access to the certificate store, the attacker can insert rogue certificates and, consequently, make the device recognize rogue remote servers as authentic.

The MAXQ1061/MAXQ1062 can store CA certificates in the internal memory and prevent anyone from modifying them. Also, the MAXQ1061/MAXQ1062 verify the authenticity of certificates prior to loading them into the internal memory. Only certificates signed by the CA can be imported to the chip, making it impossible to use rogue certificates. Once imported, those certificates can, in addition, be gated by some security conditions to further restrict usage.

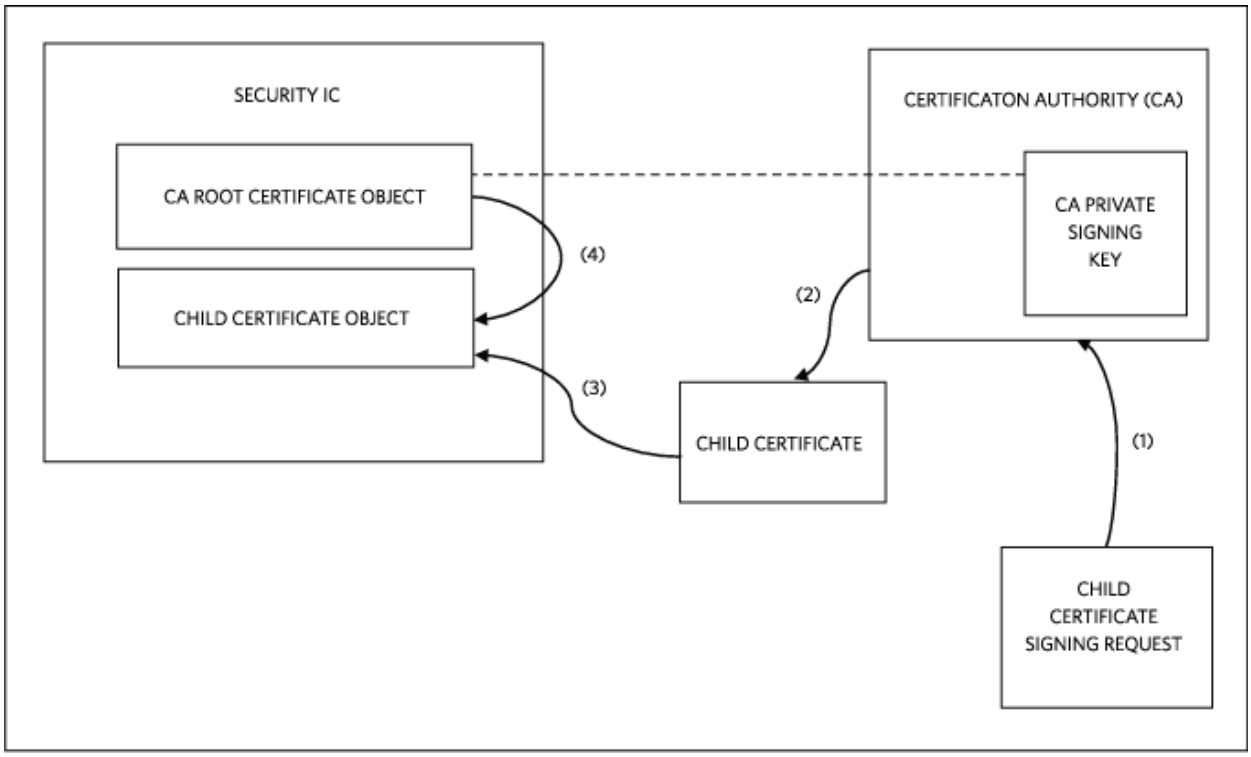

Figure 3. Child certificates verification with parent certificates.

As shown in **Figure 3**, the child certificate signing request (1) is sent to a certification authority (CA) that emits a child certificate, by using its own CA private signing key (2). The newly generated child certificate is verified onboard the MAXQ1061/MAXQ1062 thanks to the certification authority's root certificate already present in the internal memory, and then recorded into the MAXQ1061/MAXQ1062 nonvolatile storage. Note that the certification authority's root certificate (matching the CA private signing key) must be preloaded during device personalization by a trusted entity who must own the MAXQ1061/MAXQ1062 private key for importing.

Watchdog for the Host Processor

The overall reliability and security of the device can be improved when the host processor is externally monitored.

With the MAXQ1061/MAXQ1062, monitoring the behavior of the host processor can be achieved through a watchdog mechanism. In this mechanism, the host processor must periodically assert an input pin of the MAXQ1061/MAXQ1062 before a deadline. If the deadline is crossed, the MAXQ1061/MAXQ1062 assert the RESET signal of the host processor and reset the internal security context. This allows the device to exit deadlocks or stop unusual behaviors of the host processor.

When certain security errors occur, such as a failure in the secure channel between the MAXQ1061/MAXQ1062 and the host or a tamper event on the TAMPER pin, or if the MAXQ1061/MAXQ1062 reboot, the MAXQ1061/MAXQ1062 can also assert the RESET signal of the host processor. This feature guarantees that the platform remains in a secure state (reboot) if anything goes wrong.

Chips based on the TPM standard do not support such features.

Internal Secure Storage
The MAXQ1061/MAXQ1062 and the chips based on the TPM standard can either directly store small amounts of data in internal storage with access control or store data encryption/decryption keys and release them to the host processor when certain conditions are fulfilled. In the case of the latter, the host processor is in charge of encrypting/decrypting the data of the data store using the key released by the security IC. Both security ICs have a limited internal memory space which limits this use case to a few kilobytes to a few tens of kilobytes of data.

Administration
With the MAXQ1061/MAXQ1062, the administrator must use a private key for authentication. The MAXQ1061/MAXQ1062 are designed to be used on an unattended device, so the private key should be stored remotely, and the administration should be done remotely, or alternatively, during the manufacturing process. This is the preferred scenario for connected devices. The only action that is always restricted to administrators is the creation of objects in the file system. Created objects can be assigned different access rights (think about Linux's chmod command). Access to some actions on some objects can be limited and granted only to the administrator. For more information see the Designing Security Policies with Access Control section.

Administration of the chips based on the TPM standard is less suitable for IoT devices. Administration can be made by the TPM chip owner. Ownership of the chips based on the TPM standard is established by setting a password. On a PC, this password is generated by the OS and is a long and complicated one. The password is stored on a remote database (e.g., Active Directory) or is written down by the PC user. Owning a chip based on the TPM standard allows various actions to be taken such as key generation and chip reset. Note that a chip based on the TPM standard can be cleared without any password, but this operation requires a physical presence (i.e., the physical presence I/O is asserted).

The TPM presence feature exposes the chip based on the TPM standard to a local reset if someone opens the device enclosure and asserts the presence pin. The MAXQ1061/MAXQ1062 are more secure because a dynamic authentication is always required. The administrator may not have complete access to all assets of the MAXQ1061/MAXQ1062, which creates an even more stringent security policy.

Secure Channel
A secure channel is designed to protect commands and responses in transit between the security IC and the sending peer. The peer can be either the local host processor or a remote entity. The secure channel provides confidentiality, integrity, and authenticity of the commands and data in transit.

The chips based on the TPM standard can perform encrypt/decrypt sessions combined with authorization sessions. The authorization session mandates the use of a HMAC over the whole command, and also the responses, to guarantee the integrity of the command's end responses. In addition, the encrypt/decrypt session uses an AES-based algorithm to ensure the confidentiality of the commands and responses. In other words, the chip based on the TPM standard and the peer (either the local host processor or a remote peer) that communicates with the chip share the same encryption key and HMAC key. This mechanism is part of the authorization feature.

The MAXQ1061/MAXQ1062 also support this mechanism. Opening a secure channel simply calls for performing a key derivation from random numbers and a shared secret key, which is shared between the MAXQ1061/MAXQ1062 and the peer. The resulting derived session keys are used to encrypt and sign data in both directions. Encryption uses AES-CBC, and signatures use AES-CBC-MAC. The MAXQ1061/MAXQ1062 completely encrypt the command, parameters, and data in addition to the responses, making it impossible to guess what command is transmitted to the MAXQ1061/MAXQ1062. Commands and responses are cryptographically chained with each other so that swapping, removing, inserting, or replaying commands in the sequence is not possible.

On the MAXQ1061/MAXQ1062, a successful initiation of a secure channel can unlock objects and allow well defined operations.

Secure Boot

To be trustworthy, the host processor of an embedded device must run a validated firmware that is approved by the device manufacturer. The MAXQ1061/MAXQ1062 and the chips based on the TPM standard offer a secure bootstrap mechanism based on the assumptions shown in **Figure 4**.

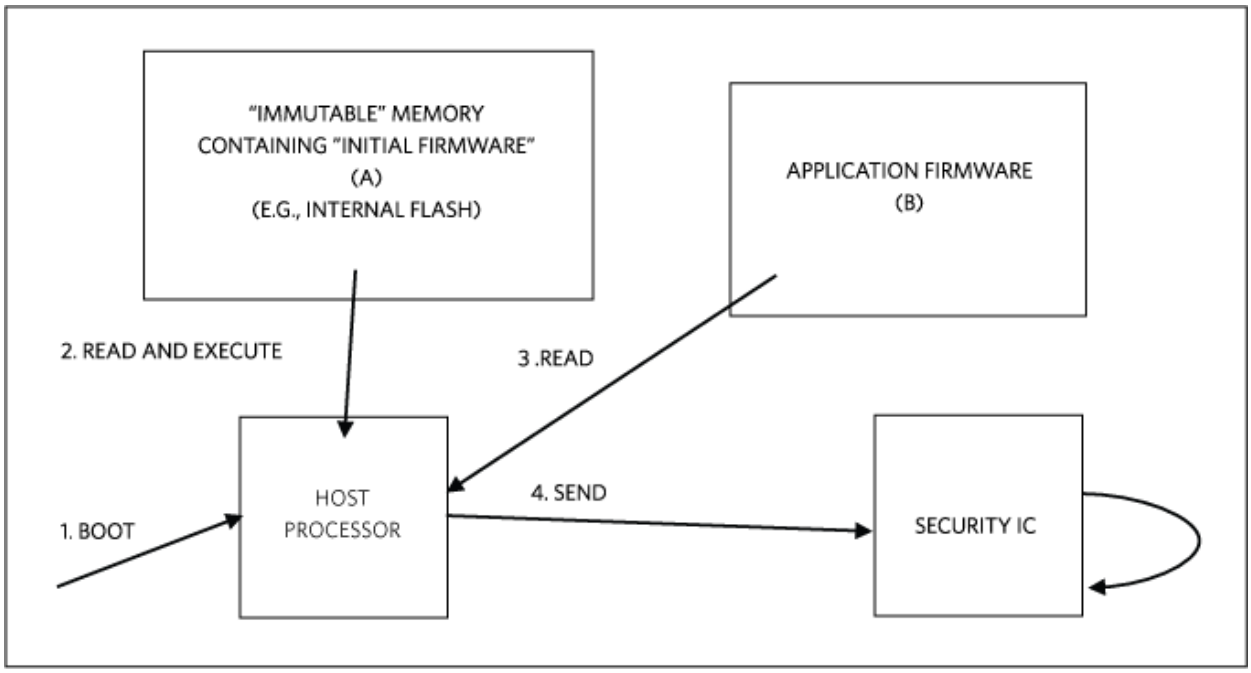

Figure 4. Secure boot.

Table 4. Description of Secure Boot

Step	Description
1	The host processor must always boot the same initial firmware located in a nonvolatile memory (A). This step is based on the following assumptions: • The location of this firmware is defined once and cannot be changed

	• The initial firmware is stored in an internal flash memory of the host processor that cannot be reprogrammed easily The host processor must provide a mechanism to make those assumptions true. The initial firmware could be stored into a flash memory that can be updated with very strict access control, and, in addition, the host processor always boots on this internal flash. This memory is the static root of trust.
2	The initial firmware executes in the host processor and performs a self-verification using the security IC.
3	Read firmware. The early firmware verifies the next stage firmware using the security IC.
4	Feed security IC with application firmware. The initial firmware sends the next stage firmware (i.e. the application firmware) to the security IC.
5	Security IC internal state modified if firmware verified successfully. The initial firmware verifies the next stage firmware using the security IC. If the verification succeeds, some security conditions in the security IC, that influence the access control, may change. For example, a private key may become available for signing data.

The technical details of the secure boot differ between the MAXQ1061/MAXQ1062 and chips based on the TPM standard. Chips based on the TPM standard verify the platform integrity (i.e., the firmware and vital data integrity) by measuring the platform. Measuring means hashing and accumulating the hash operation result in a platform configuration register (PCR). An early stage firmware executed by the host processor uses the chip based on the TPM standard to self-verify and then verify the subsequent boot stage firmware (e.g., BIOS to Disk MBR to U-Boot bootloader to Linux kernel). Each certified stage is responsible for checking the next level by feeding the binary code into the chip's PCR.

For added complexity, the chips based on the TPM standard have multiple PCRs that can be used for various purposes. Depending on the policies in place, some internal objects from the TPM-standard chips can be unlocked if the PCR values match previously recorded reference values. Note that the chips based on the TPM standard can also verify the digital signature of hashes stored in PCRs and compare PCR values to reference values.

The MAXQ1061/MAXQ1062 offer a simple and secure bootstrap mechanism by supporting only one internal hash register and requiring a signature verification. To use this mechanism, the host processor boots an initial firmware located in an immutable memory that is inherently trusted. This initial firmware feeds the MAXQ1061/MAXQ1062 with the binary code of the application that must be booted, and the MAXQ1061/MAXQ1062 hash and accumulate this code into an internal hash register. At the end of the process, the initial firmware provides the signature of the application to the MAXQ1061/MAXQ1062. The MAXQ1061/MAXQ1062 verify that the signature matches the internal hash register using a public key. A successful verification unlocks some of the MAXQ1061/MAXQ1062 internal objects for further use. Such

an object can be a private key used in an authentication mechanism that can prove to a network peer that a successful secure boot sequence has been done.

The MAXQ1061/MAXQ1062 offers a platform integrity verification that is as secure as the one the chips based on the TPM standard offer but with a simpler yet configurable security policy. The MAXQ1061/MAXQ1062 fast AES engine also offers an alternative firmware simultaneous verification and decryption that the chips based on the TPM standard do not provide.

Secure Update

A secure firmware update is more relevant than a secure boot. Both security ICs can implement a secure update. With the MAXQ1061/MAXQ1062, the implementation is very easy. The MAXQ1061/MAXQ1062 initialize with a public verification key that is used to verify the digital signature of the new firmware image. In addition, a decryption key can be rendered available when the above verification occurs, allowing decryption of the firmware. Firmware encryption is required more as high value algorithms become commonplace, even in simple embedded devices. Therefore, the algorithms need to be protected against extraction from long term storage, eavesdropping during transfer, and reverse engineering.

Thanks to the AES fast engine on the SPI bus that handles GCM mode, the MAXQ1061/MAXQ1062 can enable a fast decryption and authentication (up to 10Mbps) of the firmware. The MAXQ1061/MAXQ1062 offer either an asymmetric or symmetric-based secure boot.

Chips based on the TPM standard can also perform a secure update using fast decryption and authentication but cannot support simultaneous fast authentication and decryption.

Designing Security Policies with Access Control

The MAXQ1061/MAXQ1062 security policy is rather simple. For each life cycle state of the MAXQ1061/MAXQ1062, access control lists define which security condition is needed to allow certain actions.

As shown in **Table 5**, each empty cell defines what is allowed or not allowed depending on the object/command and the currently enabled security condition. The same access control matrix is defined (differently or not) for each life cycle state.

Table 5. Access Control Matrix

	Security Condition			
V Item	None	Host	Secure Boot	Admin
Object				
Command				

Objects in the MAXQ1061/MAXQ1062 internal memory have access conditions that can be defined at the time of creation. The object access conditions define what security conditions are needed for each possible action on the object (e.g., READ: host condition is needed, WRITE: never allowed, USE KEY: SECURE BOOT is needed).

Command access conditions are pre-defined and cannot be changed, with exception to the possibility to make the host condition mandatory for most commands. This forces all commands to be sent through a secure channel with the host processor.

The MAXQ1061/MAXQ1062 security conditions are as follows:

- The condition HOST (processor) is authenticated when a secure channel is successfully opened.
- The condition ADMIN is authenticated by a successful admin authentication command execution.
- The condition SECURE BOOT is authenticated by a successful verify boot command execution, meaning that the host firmware's digital signature has been successfully verified.

Note: Multiple security conditions can be enabled at the same time.

Explaining the whole TPM standard security policy concept is overwhelming as it is quite complex. For more information, read the book "A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security" and refer to the TCG specification. The chips based on the TPM standard achieve the same security conditions that the MAXQ1061/MAXQ1062 have despite being more complex.

Basically, there are three possible roles: USER, ADMIN, and DUP. USER is used for normal uses of the entity, the ADMIN role is used for system management tasks, and DUP, a narrowly focused role, is the only role allowed for the TPM2_Duplicate command.

The required authorization type is determined by two attributes: userWithAuth and adminWithPolicy. These attributes are either set explicitly, such as at object creation time for objects, or determined by other means for certain types of entities handled by the chip based on the TPM standard.

For the chips based on the TPM standard, the three types of authorizations are as follows:

- Password. The sender of the command sends a simple password for authorization to perform the command.
- HMAC. The password is used as a basis for the HMAC key. The HMAC is computed over the command to guarantee the integrity of the command and that the command is sent by someone who knows the password (without revealing it). Random numbers are used to avoid getting twice the same HMAC value when sending twice the exact same command, which is also known as replay attack protection.
- Policy sessions. Policies very finely define when actions are authorized, by including the values of certain states (e.g., PCR values, sequences of commands, object values, time values), defining very complex schemes, and even mandating a specific sequence of commands.

Overall, the MAXQ1061/MAXQ1062 security policy is flexible enough to use while being easier to understand and implement.

Life Cycle State

The MAXQ1061/MAXQ1062 have three major life cycle states based on a strictly monotonic life cycle counter. Each state provides a different access control policy. A fourth state named terminated can be used to decommission the part, and hence the device, completely and permanently. The transitioning to this state is restricted to the administrator but can also be permanently disabled if needed. Transition between states happens only when the administrator sends a dedicated command. This can be useful during manufacturing and personalization of the device to grant or deny access to some objects, depending on the manufacturing step.

The chips based on the TPM standard can go in both directions. The chip based on the TPM standard is

enabled when ownership is established. When the owner sends an erase command, the chip is erased. Although the presence I/O must be asserted, erasing the chip based on the TPM standard cannot be prevented, and this can be a security concern in some cases.

Monotonic Counters

The TPM standard features 64-bit counters that only increment. Counters can be used to build complex security policies. For example, a complex security policy requires a key that expires after a certain number of uses. The MAXQ1061/MAXQ1062 feature 32-bit counters with two types: incrementing only and decrementing only. Contrary to the chips based on the TPM standard, the MAXQ1061/MAXQ1062 counters have no particular side effects.

Time Management

The chips based on the TPM standard have an integrated timer for secure time stamping and the expiration of some lockouts, keys, or certificates. The timer stops running when the chip based on the TPM standard is powered off. To overcome this limitation, the chips have a boot counter that increments each time the chip boots. Otherwise, the timer can be securely synchronized to an external clock source after each reset. The TPM standard offers the capability to sign using the current time and boot counter values or to bind data to the timer. For example, one can sign the current timestamp and a ticket that proves that an operation has been performed in order to certify that the operation was performed at a certain time (i.e., signing a contract). Making a key that is usable only until the next reboot is another example. The MAXQ1061/MAXQ1062 do not have time or reboot counters, but by computing a digital signature of an RTC input from the host microcontroller, the MAXQ1061/MAXQ1062 can be used to provide a secure timestamp. Because the TPM standard requires external clock synchronization, the constraints for the MAXQ1061/MAXQ1062 are similar to the ones for chips based on the TPM standard.

Backup/Restore and Key Migration

Unlike the chips based on the TPM standard, the MAXQ1061/MAXQ1062 cannot export secret or private keys but only public keys or certificates. However, importation of secret or public key pairs is available, and the key encryption during the importation is possible by using the secure channel only. Chips based on the TPM standard can import/export keys encrypted with a transport key. This type of key backup/restoration mechanism is useful on a computer when the user needs to migrate data onto another machine in case of replacement. On embedded devices, this may not be very useful. Therefore, the MAXQ1061/MAXQ1062 functionality is enough.

MAXQ1061/MAXQ1062 Personalization and Provisioning Services

One of the most predominant applications of the MAXQ1061/MAXQ1062 is to enable an offline public key infrastructure. Devices must be pre-provisioned with certification authority certificates to be able to either authenticate other peers or to be authenticated by others while having no access to online certificate servers. See Trust Your Digital See Trust Your Digital.

Chips based on the TPM standard are delivered with a pair of keys known as the endorsement key (EK) pair, which is unique to each IC. The EK pair consists of a public key (Ekpub), which is certified by the chips manufacturer, and a private key (Ekpriv). The purpose of the EK pair is to verify that the chip is authentic. During manufacturing, the EK pair is generated inside the chip. The Ekpriv never leaves the chip memory, but the Ekpub is read out and then certified by the manufacturer, and the resulting certificate is loaded back to the nonvolatile memory storage of the chip. After delivery, customers can verify the chip's authenticity by using the Ekpub certificate to verify a digital signature computed by the chip's Ekpriv. The manufacturer's root certificate is available online so that customers can verify the

authenticity of the chips Ekpub certificate before using it to verify the signature. Note that customers can replace the initial EK pair to avoid being traced back to the manufacturer. While manufacturers of chips based on the TPM standard may offer personalization services, it is usually up to the customer to perform the whole chip's provisioning.

By default, the MAXQ1061/MAXQ1062 ICs are shipped with a default administrator authentication public key and a default key for importation. However, upon request, the MAXQ1061/MAXQ1062 ICs can be delivered with some pre-personalized objects including additional administrator public keys and key importation public keys, additional pairs of keys and certificates, or any other kind of object. This personalization service allows customers to avoid the burden of handling keys and certificates or generating certificates in their manufacturing facilities. Instead, the MAXQ1061/MAXQ1062 ICs are mounted on PCBs with all the information needed to bootstrap the final device. The MAXQ1061/MAXQ1062 can be shipped to customers with a unique pair of keys already present in the internal memory, where the keys are generated in place, and the private key never leaves the internal storage. A certificate of the public key can also be generated at Maxim's manufacturing facilities using a Hardware Security Module (HSM). The certification authority can be from Maxim, a customer, or a third party.

The MAXQ1061/MAXQ1062 offer full flexibility in terms of keys and certificates that are provisioned because there are no pre-defined roles for keys or objects.

Security Limitations of Security ICs

The MAXQ1061/MAXQ1062 can overcome the inherent limitations of a bi-chip design. The inherent limitations are as follows:

- The capability for an intruder to tamper with the communication link between the host processor and the security IC.
- The capability for an intruder to replace the initial firmware and skip the secure boot stage but still feed the security IC with the genuine firmware to get access to the assets locked by the SECURE BOOT condition. For example, see Ledger Addresses Man in the Middle Attack That Threatens Millions of Hardware Wallets.

Chips based on the TPM standard are completely unarmed against such attacks since they rely on the assumption that only remote attacks are possible.

The MAXQ1061/MAXQ1062 mitigate attacks by using a tamper input pin that monitors the device enclosure integrity. As soon as the enclosure is opened, vital assets can be erased from the MAXQ1061/MAXQ1062 internal secure storage, and then the device is prevented from connecting to the network with an insecure firmware or state.

However, implementing the tamper detection at the system level requires the addition of switches and/or special tamper-sensitive artifacts. Plus, a permanent power supply for the MAXQ1061/MAXQ1062 should be added so that it can actively erase the assets at stake. The extra cost for this design implementation may be worthwhile.

Features Not Implemented by the MAXQ1061/MAXQ1062

Due to the initial design of chips based on the TPM standard, the chips can implement many PC-oriented use cases that are not especially relevant in small embedded devices.

- PKCS#11 or Microsoft CAPI engine. These cryptographic service providers can leverage basic TPM standard functionalities to provide more secure cryptographic key storage and algorithms to the upper software layers. The service providers support less functions than the TCG Software Stack. The MAXQ1061/MAXQ1062 can be implemented in such engines but are not really designed for that purpose.
- Chips based on the TPM standard offer digital rights management and software license verification. As an example for digital rights management, when a video is purchased for a limited time of 30 days, then the video decryption key expires after 30 days. Software license verification can be achieved by verifying that the identity of the platform matches the one registered in the license and also by verifying that the license has not expired.
- Remote anonymous attestation. While this can be important during transactions involving users, this operation proves the security properties of a platform (like platform integrity) without revealing its identity. Remove anonymous attestation uses a specific algorithm (DAA) that is not implemented in the MAXQ1061/MAXQ1062. This advanced concept is powerful but complex and might not be that useful in embedded applications, as a simple digital signature scheme is sufficient.
- Password manager. Similar to the disk-encryption use case, a password manager needs proof of integrity of the platform before releasing the password-store decryption key. It also requires the entry of the owner's master passphrase for the password wallet or a successful matching fingerprint. In the context of IoT, the lack of a password manager is not seen as a real limitation because passwords are seldom used.

Conclusion

Even if there are some differences regarding the list of supported cryptographic algorithm variants, this is usually not a decision factor for customers unless very specific cryptographic requirements exist. The TPM standard and the chips based on the TPM standard are very complex to use and understand and require a relatively larger host-side software stack. In addition, it is also unclear if manufacturers of TPM-compliant chips are willing to offer personalization services such as the pre-loading of unique certificates in each chip.

The MAXQ1061/MAXQ1062 are perfectly adequate for IoT security scenarios and requirements. Unless special requirements regarding access control policy or certification must be fulfilled, the MAXQ1061/MAXQ1062 are always a better choice than the chips based on the TPM standard thanks to the simplicity of the security concepts, the verifiability of the platform's security, the reduced command set, the unique features such as the host-processor watchdog and reset control, the device enclosure tamper detection feature, and the integrated TLS protocol processing. The MAXQ1061/MAXQ1062 are easy to implement and require a small footprint on the host processor memories.

Related Parts		
MAXQ1061	DeepCover Cryptographic Controller for Embedded Devices	Free Samples
MAXQ1062	DeepCover Cryptographic Controller for Embedded Devices	Free Samples

More Information
For Technical Support: https://www.maximintegrated.com/en/support
For Samples: https://www.maximintegrated.com/en/samples
Other Questions and Comments: https://www.maximintegrated.com/en/contact

Application Note 6762: https://www.maximintegrated.com/en/an6762
APPLICATION NOTE 6762, AN6762, AN 6762, APP6762, Appnote6762, Appnote 6762