Protecting People, Data and Revenue with Security-Optimized Embedded Designs – Part II

Design Practices for Secure, Cost-Effective Embedded Products

Atmel White Paper

By: Todd Whitford, Product Marketing Manager, Atmel Corporation and Steve Jarmusz, Applications Engineering Manager, Atmel Corporation

Abstract

If you're involved with embedded systems, ignoring security is not an option. The products you design and manufacture today will enter a hostile environment, crawling with all manner of cyber-thieves, industrial spies and hackers intent on attacking your devices - and the customers who use them. The first installment of this article provided an overview of the security threats faced by the electronics community, their potential impact on a manufacturer's profits, the security of its customers, and ability to compete in the global market. After a brief recap, this concluding installment will provide a more detailed look at some of the major threats and use cases which illustrate cost-effective countermeasures which can be tailored for nearly every application.

The Security You Don't Pay For Can Cost You Plenty

Protecting a system against intrusion and data theft requires it to provide appropriate levels of both design security (protecting a system's software and other critical details about its design) and data security (protecting the information stored in, or transported by the system). Failing to do so can expose your system to various forms of cyber-theft which seek to extract the information stored or transported by an embedded system. In many cases, the information contained in leading-edge products is pilfered by hackers serving second-tier manufacturers (often located offshore) who are employed to produce copycat products --or as an inexpensive alternative to doing their own R&D. The resulting over-builds, cloned products, and unauthorized knock-offs of proprietary systems, peripherals and accessories can erode or eliminate a manufacturer's profits.

Poorly-secured products also put your customers at risk. Weaknesses in network connections can be easily exploited to intercept the sensitive data they carry. They can also be used to inject malicious code during routine software upgrades. Once inside your system, the new code can turn it into a convenient point of entry to your customer's (or your) network that can be used to gain access to sensitive consumer and corporate data.

Security Architecture Options

Hardened systems which protect their own secrets as well as the information they are entrusted with have traditionally relied on specially-designed processors and peripheral components. These devices are "hardened" with a combination of logic and physical mechanisms to deny access to the data they store. While they cost more and have often delivered lower performance, secure processing chip sets were the only way to insure high levels of design and data security.

Recently however, an alternative architecture has emerged which uses a single inexpensive device known as a secure key manager. The secure key manager's non-volatile memory is used to store keys, passwords, and other sensitive data which cannot be extracted from the device. Its on-board security logic and random number generator protect against logical attacks on the data transmitted between the

device and the system. This mechanism can also be used for authentication of any device attempting to communicate with the system. To protect against physical attacks on the device itself and the information it contains, the key manager is equipped with an extensive set of logical and physical defensive mechanisms.

Using a dedicated device to perform secure key storage and authentication can be a highly cost-effective alternative to a fully-hardened embedded system in many security-critical applications. These devices can be paired with virtually any MCU to keep the system's sensitive data and security-related operations within the confines of a single, low-cost hardened device. The resulting system is highly-resistant to the attacks commonly used to steal sensitive data, software and other types of intellectual property (IP) at a much lower solution cost than a fully-hardened system.

The products you design and manufacture today will enter a hostile environment, crawling with all manner of cyber-thieves, industrial spies and hackers, intent on attacking your devices - and the customers who use them.

The Atmel CryptoAuthentication™ ATSHA204

For the remainder of this paper we'll use the Atmel CryptoAuthentication ATSHA204 to illustrate how these devices work and how they can be used to provide both data and design security in embedded applications (Fig.1). The ATSHA204 is an advanced secure key storage and authentication device that includes 4.5Kb of EEPROM. Its secure memory can be used to store multiple keys, perform miscellaneous read/write or read-only memory operations for managing passwords or secret data, as well as consumption tracking information. Access to the various sections of memory can be restricted in a variety of ways and then the configuration locked to prevent changes. The device features a wide array of defensive mechanisms specifically designed to prevent physical attacks on the device itself or logical attacks on the data transmitted between the device and the system.

Fig.1: - The Atmel ATSHA204 advanced secure key storage and authentication device is a cost-effective way to harden embedded systems against IP theft and other cyber attacks.

Each ATSHA204 device ships with a guaranteed unique 72-bit serial number. Using the cryptographic protocols supported by the device, a host system or remote server can prove that the serial number is both authentic and not a copy. The ATSHA204 can also generate high-quality random numbers and employ them for any purpose, including use by the device's own crypto protocols. The technologies' flexible command set makes it easy and cost-effective to use in applications requiring anti-counterfeiting and anti-cloning protection, firmware IP protection and validation, secure data storage, user password checking and session key exchange.

In the next section, we'll explore several of these use models using the Atmel ATSHA204 CryptoAuthentication device.

- **Consumable Authentication/Anti-Counterfeiting and Anti-Cloning**

  In this use model, the ATSHA204 device is used to validate that a removable, replaceable, or consumable client product is authentic. The following scenario illustrates how a simple and inexpensive crypto device such as the ATSHA204 makes it possible to ensure that only authentic ink cartridges, air filters, medical consumables, etc… are accepted and permitted to function with authorized OEM systems.

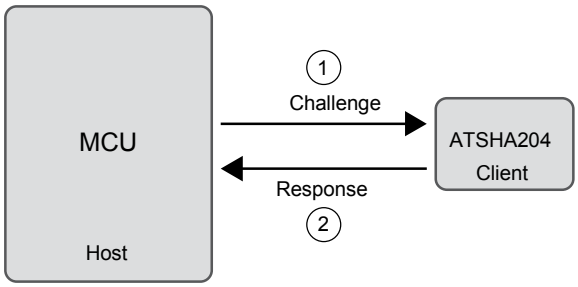

Fig.2: - Atmel's ATSHA204 can be used to authenticate an embedded system's consumable elements, or protect against production of unauthorized copies of the system itself.

For anti-counterfeiting and/or anti-cloning applications, the device is embedded in the consumable (Client). When the consumable is installed or first used in the system (Host) it's sent a challenge from the Host in the form of either a fixed or random value. Once the Client receives the challenge, its ATSHA204 security device calculates a response value and sends it back to the Host where it is compared with the expected value. Only a consumable that provides the expected response can be used in the system. In addition, the onboard ATSHA204 can be used to track and/or limit usage. This is particularly useful in systems where tracking or limiting the use of the consumable is important to ensuring product reliability and, in some cases, consumer safety. This type of consumable authentication system is an inexpensive way to insure that both your customers and your revenues are protected.

Using a dedicated device to perform secure key storage and authentication can be a highly cost-effective alternative to a fully-hardened embedded system in all but the most security-critical applications.

❖ **Accessory Authentication**

A similar approach can be used to make sure your system will only work with manufacturer-authorized plug-ins, cables, batteries, or other accessories. In some cases, the primary damage caused by knock-off accessories such as power supplies, speakers, or specialized cables is lost

revenue. In other cases, an unauthorized accessory such as a battery, surgical instrument, or medical device which does not fully meet OEM requirements has the potential for causing serious harm.

In these applications, also including the ATSHA204 device in the host allows it to generate a Random Challenge for the client (accessory) and evaluate its response. In addition, since the host's challenge is generated internally using its ATSHA204, even the host processor is unaware of its response. Isolating the processor from the security function enables the use of an inexpensive non-security-hardened processor without providing a weak point through which an attacker can extract the system's secrets.

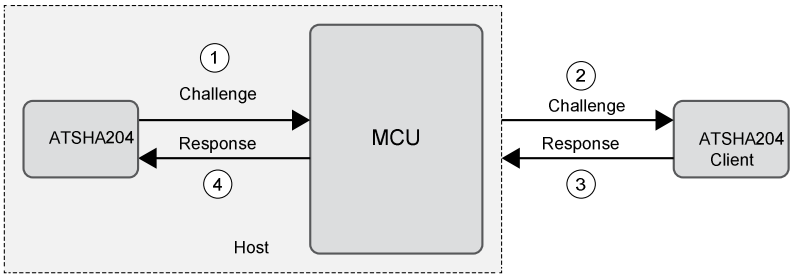

Fig.3: - A secure key storage device embedded in an accessory permits verification of its authenticity by the host system. Including a secure key storage device on the host as well significantly improves system-level security.

- **Firmware IP Protection**

Protecting flash-resident firmware, which may contain proprietary algorithms or other IP, against theft can be accomplished by encrypting it before it is stored. A secure key storage device such as the ATSHA204 manages the keys which the system MCU uses to decrypt the firmware and other information stored in flash memory.

If needed, the ATSHA204 can also be used to implement other measures which provide additional layers of protection against software theft. These include:

- **Multiple challenge-response pairs**

When you use multiple challenge-response pairs, the system will choose a set of challenge/response pairs based on some algorithm in the system code. This could be a function call to the c library rand() or a fibonchi lfsr. The number of challenge/response pairs is limited by the amount of space that a given system has to store the support code and challenge/response pairs. In addition, this scenario could be made more complex by offsetting where the challenge and its corresponding response are or where it is held in memory (i.e. the challenge could be held in array 5, while the response could be held in array 23).

- **Chaining challenge-responses**

In the chaining Challenge Response Technique, each response from the ATSHA204 can be fed back out as the new challenge. At some point the response would be evaluated and checked so that the authentication is verified successfully. By not evaluating the response each time the system gets the response from the client, the chain could execute a specified number of rounds without triggering a negative effect. If a hacker were monitoring the bus and failed the authentication check, they would not know which challenge/response was invalid.

- **Code Misdirection**

Code misdirection is the addition of code in the equation that obfuscates to some degree the code path that is being executed, thereby making it harder for would-be hackers to clone a device. When a function pointer is declared, a check is done within a local function. Once the answer is received the function pointer is set to null. This process makes it significantly harder to de-compile the source code required to clone a device. Code misdirection could also be used to point to code that causes severe penalties if the response to a given challenge is incorrect, such as pointing to an infinite loop or, if desired, code that does something destructive.

- **Move the Challenge to TempKey**

In this example technique, a challenge could be stored in a reserved 32-byte register of the ATSHA204. At some point much later, the MAC command could be run on the stored challenge and the response then could be sent back to the system. This additional step makes it much harder for a would-be hacker to pair a given challenge to that response.

- **Rolled Key Mechanism**

Instead of using a “static” key in the authentication calculation, the rolled key function in the ATSHA204 adds security by changing the key value used in the calculation. This is accomplished by combining some offset values and creating a new key. The offset value could be derived from a variety of sources including the unit’s serial number, a time stamp or an internally-generated random number. This new key would permanently remove the original key. After the key has been changed, there is no way to recover the original key. Instead of using the challenge and response mechanism as the primary source of protection, the keys themselves now become that protection.

- **Preventing Cloning, Over-Builds and other Types of Firmware Theft**

In highly-competitive markets, a product’s firmware is often one of its key differentiators and must be protected against exposure and resulting theft for use in grey market knock-off products. In these applications, challenge-response functionality, diversified key schemes, rolling keys, and other protections are implemented to thwart would-be thieves.

- **Firmware Validation / Secure Boot**

If an embedded device’s operating program is stored in an external Flash device, it's very hard to prevent its contents from being copied and modified to run a fraudulent program. The ATSHA204’s unique key management capabilities can be used to implement a secure boot function which ensures that only the manufacturer’s authenticated firmware can be run on the system.

To implement secure boot, a validating code or signature is stored in flash along with the boot code. The signature is created at the time of manufacture or code update and is the output of a cryptographic hash function, an irreversible algorithm which “condenses” the boot code into a compact, but unique datagram, typically 32-128 bytes in length. Figure 4 below illustrates how the calculation and key storage is offloaded into a hardware key management device (ATSHA204).

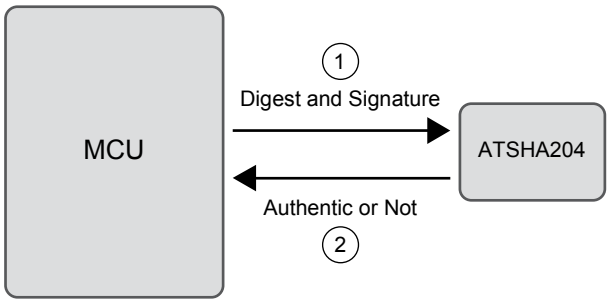

Fig.4: - The Atmel ATSHA204 can be used to protect against firmware tampering and support the authentication functions required for a secure boot procedure.

At system startup, part of the boot sequence requires the security device to verify the signature accompanying the boot code. Only if the verification succeeds will the operating program be executed, and allow the system to operate in the normal way. Even modifying the operating program by a single bit will require a new validating signature. Both code images and downloaded media files can be stored using an encrypt key which can be usable only on a single authenticated system.

Adding Atmel's ATSHA204 secure key storage device to your embedded application can provide the key and data management, as well as other critical security capabilities needed to meet most security requirements without heavy impact to its development schedule or cost.

❖ **Network Security**

Wireless transmission devices must verify each node prior to allowing access to the network. ATSHA204 devices in each radio node (Client), allows the transmitting node (Host) to verify that it is communicating with valid network nodes before transmitting important commands or information. Adding an ATSHA204 device to each node provides safe storage of keys and/or data. Figure 5 illustrates a configuration that utilizes two ATSHA204 devices in a radio network.

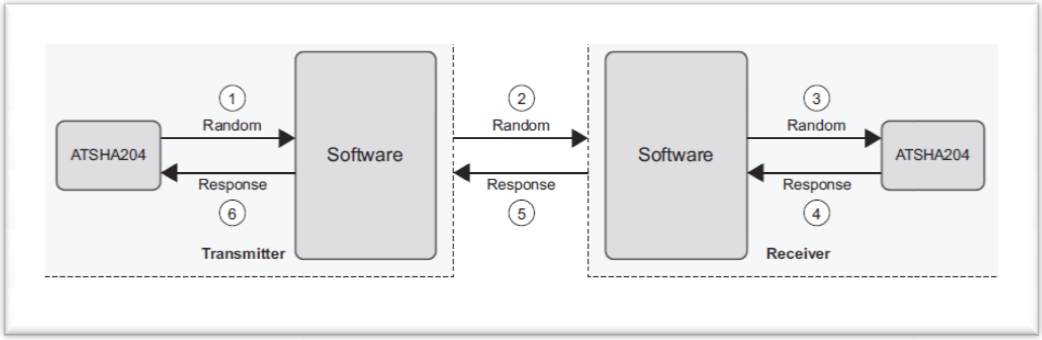

Fig.5: - Atmel's ATSHA204 can perform the challenge/response exchanges required in applications where authentication of both host and client are required before establishing a connection.

• Secure Session Key Exchange

In order to manage a confidential communications channel or an encrypted download, a host system and its client exchange stream encryption keys in a secure manner. The keys are then used by an encryption/decryption engine in the system microprocessor to support symmetric security algorithms such as AES. The ATSHA204 can facilitate secure key exchanges by using the unique response it produces as a key to the symmetric encryption algorithm.

A random number is used in the encryption key's generation process. This can be supplied by the ATSHA204's on-chip random number generator. The key exchange is then performed by sending a random challenge to the Host ATSHA204, which in turn generates a response. The host's response is used as a session-key for encrypting the message. The message and the random number are then sent to the client ATSHA204. On the client side, the random number is fed into the ATSHA204 device to generate the response which is used as the key to decrypt the

message. It should be noted that the root key is the same on both the host and client and never leaves the ATSHA204 device.

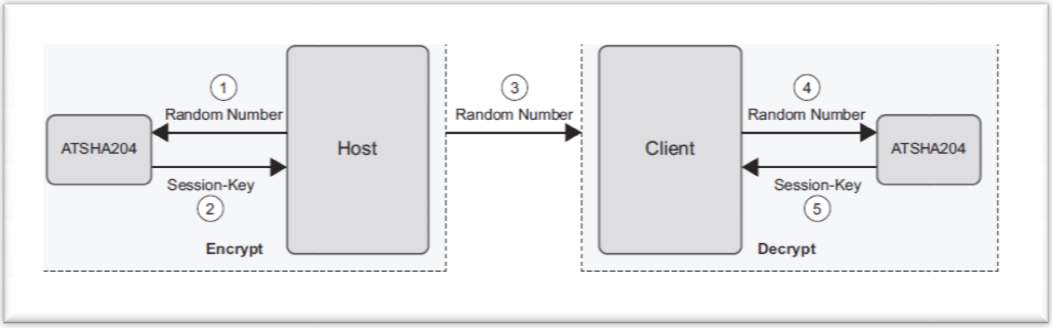

Fig.6: - In secure session key exchanges, the ATSHA204 offloads the encrypt/decrypt functions from the system MCU as well as providing secure storage for all encryption keys and related variables.

Atmel Makes Security Easy – And Effective

The ATSHA204 is part of an integrated set of security solutions created by Atmel to make development simple and reduce time to market. It comes with a rich library of basic security functions which are fully-integrated into Atmel's Studio6 development environment so even non-experts can create fully-hardened secure designs. Customers with short development schedules may choose to implement one of the pre-developed turnkey solutions available for many popular applications. This reduces development time and simplifies the configuration that's normally required, and can be paired with all Atmel MCUs as well as most other common processors.

Additionally, programming and locking the secrets and/or data in the security device prior to deployment is a necessity. Atmel has numerous personalization options available, enabling fast time-to-market.

Summary

You and your customers rely on embedded security technologies to protect your data, IP and revenue – and your customers. Some of the considerations which will influence the security measures you incorporate in your products include:

- Protecting Your Revenue - Securing IP to prevents over-builds, hardware and software clones and other violations of contract manufacturer agreements.

- Protecting Your Competitive Advantages – Prevent reverse-engineering of proprietary features and technologies which differentiate your product from its competition helps preserve your competitive lead in an unrelenting market.

- Protecting Your Customers - Your customers must be assured that your product will protect their networks, data and revenues against intrusion or theft.

- Protecting your Brand Image – You cannot afford to let cheap, poorly performing look-alike products tarnish your product's reputation and perceived market value.

- Reducing Liability Exposure – Prevent potential legal issues arising from copycat products, unauthorized accessories and consumables.

In many cases, adding Atmel's ATSHA204 secure key storage device to your embedded application will provide the key and data management, as well as other critical security functions to meet these challenges without heavy impact to its development schedule or cost. In highly demanding applications which require exceptionally high security and performance, the ATSHA204 provides a strong foundation upon which to build an advanced secure computing architecture. Regardless of the solution you choose, Atmel's integrated solution set and development tools dramatically reduce the time and resources required to protect both you and your customers.

For more information about Atmel's ATSHA204 CryptoAuthentication device visit http://www.atmel.com/devices/ATSHA204.aspx.

For more information about Atmel's complete line of security solutions, visit http://www.atmel.com/products/other/default.aspx.

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370