ATMEL WHITE PAPER

Securing Host Operations with a Dedicated Cryptographic IC - CryptoCompanion™

By Kerry Maletsky, Business Unit Director Crypto Products

Summary

There is a growing need for strong hardware security devices in digital systems today. The level of intellectual property or engineering expertise contained within a product continues to grow and as a result it becomes increasingly imperative for the manufacturer to prevent competitors from fraudulently using that IP to obtain a portion of the revenue stream without the associated development cost.

Well known examples include those systems with replaceable or consumable items such as batteries or test strips in a portable medical analyzer. Simply using a barcode or hologram label on the authorized product doesn't help because neither the host system nor the consumer has any way to tell if it has simply been photocopied.

As systems come to depend more and more on firmware and/or software, it becomes easier for less skilled adversaries to obtain the code to clone the entire system, modify it to permit fraudulent or unauthorized use of the system or digital content or publish it. In the past, systems often included hard-to-copy ASICs, but these are less and less used.

CryptoMemory® can protect this firmware or code by carrying the keys that are used to encrypt the code when stored, by containing parts of the code itself and by containing key values necessary for proper system operation. www.cryptomemory.com

Table of Contents

Host Side Implementation

To validate the authenticity of a consumable or replaceable item (a.k.a. client) connected to a system (a.k.a. host), there must be some secret information stored in both the client and the host. If the host can determine that the data stored in the client matches what it has stored in memory, then the client is considered to be authentic.

But to do this in a way that an adversary can't figure out the secrets requires significant cryptographic capability and expertise. Just sending the secrets back and forth will certainly not work, so the secrets have to be fully encrypted whenever they are transmitted. Further, just encrypting them won't work, because then an adversary could just record a transaction and "replay" it later when attached to the fraudulent client device.

Atmel's CryptoMemory and CryptoRF® chips provide the ideal solution for the client side. They include storage for the secrets and a built-in encryption engine. In use, the host sends a random number to the CryptoMemory chip, which cryptographically combines that with the secret and sends it back. The host makes the same computation and if the two results agree, the client must be authentic.

There are three key requirements of the host:

1. Securely store the secret
2. Generate random numbers
3. Compute the same cryptographic function as CryptoMemory

Not surprisingly, each of these three tasks is very difficult to do very well.

Most systems include some sort of nonvolatile memory for configuration, program storage or user data. But memory chips are commodity devices with very standard interfaces – it's easy for any attacker to pry the chip off the board and read the contents with very standard equipment.

Typical solutions for secure storage using standard components involve (among other solutions) some of the following:

1. Store the secret in 'locked' memory on the microprocessor. It's only a little harder to read these areas of memory since the parts are not built to defend against physical attack.
2. Store the secret distributed in many places in the NV memory. The software builds up the actual secret by reading a little here and a little there and assembling the final secret in SRAM. One can obtain the secret by reverse compiling the firmware to see what it gets when, then reading that information from the NV memory. In some cases, it is easier to attach a debugger to the system being attacked and to just dump out SRAM.
3. Store the secret in nonvolatile memory in an encrypted form. This is really no solution because there is no place to put the encryption key that is safe.
4. Require that the user enter some data (i.e. pass phrase) that contains the secret. This appears safe but is easily thwarted by any kind of malware that records activity on the keys for later retrieval by the attacker. And in some cases it is not in the users best interest to keep the password secret.

5. Secure storage devices are available. One form of these is known as a High Security Module (HSM). For most high volume systems the cost of such solutions is impractical.

Random numbers would seem to be easy to generate, but in fact are the sole source of weakness for a number of well known and successful attacks on security systems. Typical host solutions:

1. Use a collection of semi-random physical properties that are combined. Examples might be seek time on a disk, clock time, temperature, key stroke delay or value. While these appear random, in actuality there are relatively few possibilities for each and an attacker can exploit this limited range quite easily.
2. Store a random seed in NV memory and use one of the well known Pseudo Random Number Generator cryptographic algorithms to generate a stream of random number generator (RNG) values. This requires not only significant cryptographic expertise but it falls apart entirely if the seed cannot be stored in such a way that it can never be read – which is the same unsolved problem as above.

Modern cryptographic algorithms are very strong, but they are also very unforgiving. If a single bit is wrong in one of the intermediate values, all the bits will likely be wrong in the output value. This property makes it very difficult to debug cryptographic software. Also, if the algorithm is not very carefully implemented, then an attacker can observe things like timing, error conditions or other side channel information to make his/her job significantly easier.

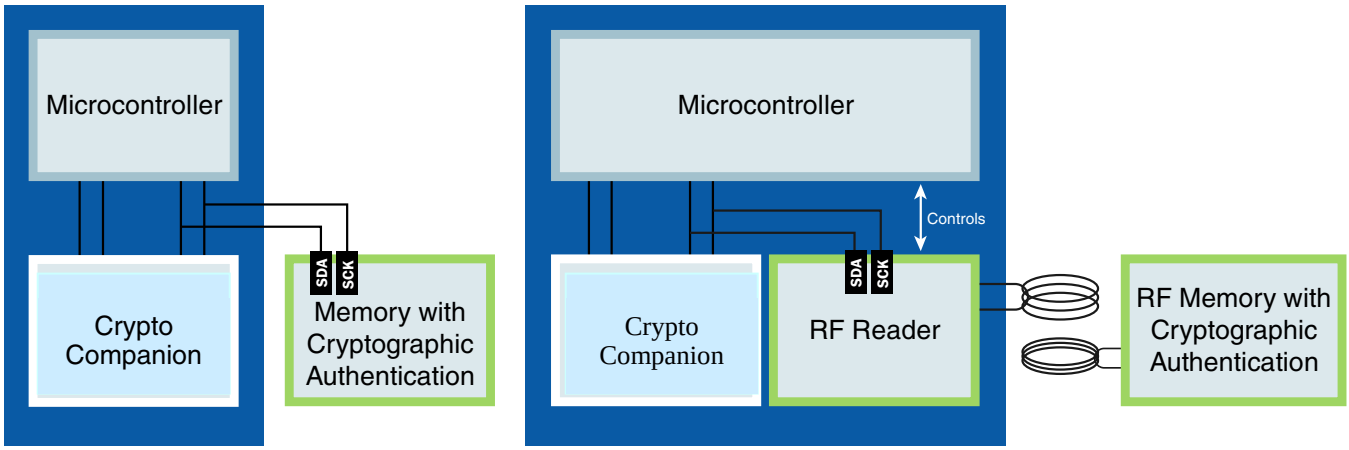
Standard libraries are available but their quality is usually unknown. Very high quality certified and tested crypto software is available, but the budget may not accept the license terms. Understanding all the specifications to “roll your own” can be time consuming when testing is included.

System designers have performed these three functions with varying degrees of success for some time. But now there is another, much better, option. The CryptoCompanion chip from Atmel provides a way to securely and easily perform all three of these host operations directly within the secure confines of a dedicated cryptographic IC.

To simplify the task even further, Atmel provides a number of different evaluation and development kits that include both CryptoMemory and CryptoCompanion chips in a single environment. Atmel also provides complete driver information for the chips compiled for a number of different microprocessors.

Crypto Companion

Figure 1. Cryptographic Chart



Key Diversification

Because many consumable items are low cost they often use just the CryptoMemory or CryptoRF chip in a tag or module form factor. This form factor is more readily accessible to a potential adversary who might try to crack it to find its secrets. On the host side, the CryptoCompanion chip is usually soldered down to a board with more expensive components and attacks are more difficult.

The recommended CryptoMemory/CryptoRF protocol takes advantage of this difference in attack potential by "diversifying" the secrets stored on CryptoMemory/CryptoRF. The companion chip stores a single "root" secret, while each CryptoMemory/CryptRF chip stores, in nonvolatile memory, a unique value generated from a combination of the CryptoMemory/CryptoRF serial number and the root secret. The serial number is freely readable, so when an item is to be authenticated, the companion chip can derive that value on the fly from the root secret and the item serial number.

What does a designer have to do?

The usual authentication method is for both the crypto chip on the CryptoMemory/CryptoRF client and the host to compute the same function on a set of random numbers and a shared secret value. If both sides generate the same internal result, then the authentication is considered valid.

There are two cryptographic algorithms that must be implemented on the host: the first generates the secret stored on the client CryptoMemory/CryptoRF from the serial number and root secret, and then the second matches the cryptographic run on CryptoMemory/CryptoRF during authentication. Any error in this implementation, even by a single bit, will cause the entire authentication to fail.

Furthermore, most successful attacks on cryptographic systems depend less on cracking the base algorithm and more on exploiting weaknesses in the implementation or the overall protocol.

The CryptoCompanion chip solves both of these potential problems by implementing both algorithms and the entire protocol completely in hardware. It is fully tested and guaranteed to work properly with all CryptoMemory/CryptoRF chips. So the system designer doesn't have to design the software and also doesn't have to test it.

The companion chip also provides a very useful high endurance nonvolatile counter that can be used by the system to count events much like an odometer on a car; the CryptoCompanion chip has four separate counters. These counters can be used to uniquely tag transactions, generate IV values for encryption, etc.

The counters can never be reset, nor can the value of the counter ever decrease – and the maximum count value is 64 million. Implementing such a counter using software and normal nonvolatile memories can be cumbersome and require a lot of space.

What else does the CryptoCompanion chip do to protect the secrets stored within it?

Based on our long experience within the digital security space, we have designed the companion chip to make software attacks on the keys more or less impossible and hardware attacks on the chip especially hard.

Key internal areas are protected with metal layers that prevent probing, internal memories are encrypted, and the chip includes tamper circuits designed to detect if it is being operated out of its normal range. Several delay circuits are included to prevent exhaustive dictionary attacks on the secrets or on authentications with the CryptoMemory/CryptoRF client chips.

How does the CryptoCompanion chip's random number generator help security?

It's very difficult to generate a true random number. Many systems have tried to do this using clocks, key stroke timing, transmission delays, etc. Unfortunately, these methods almost always result in some kind of predictable behavior which allows attacks. The literature is full of successful exploits against these systems.

If the random challenge going into the authentication protocol could be repeated, then an attacker could record a session and then replay it later to make an authentication appear valid for a fraudulent device. And some modes of AES and DES, such as counter mode, absolutely require unique seeds (also known as IVs) to guarantee the confidentiality of the data. System designers can use this capability of the CryptoCompanion chip to support other crypto protocols as well as the CryptoMemory/CryptoRF protocol.

The companion chip RNG uses a multi-stage hardware protocol based around an internal hardware noise source, hardware pseudo-randomizer and the full FIPS 140 PRNG. This world class implementation increases the designers' confidence in the numbers returned by the CrptoCompanion chip.

What algorithms are used?

The companion chip uses industry standard algorithms to derive the CryptoMemory/CryptoRF key from the root secret and the serial number. CryptoCompanion uses SHA-1 in the key derivation algorithm, which provides excellent protection for the root secret. No practical attacks on the algorithm are known which might allow an adversary to retrieve the root secret from a derived CryptoMemory/CryptoRF secret.

The companion chip also uses an industry standard FIPS procedure to generate high quality random numbers. Unpredictable random numbers are essential to prevent replay attacks and are required for proper use of several modes of standard encryption algorithms like AES.

What are the sizes of the secrets stored in the CryptoCompanion chip?

The CryptoCompanion chip can store up to 16 root keys for CryptoMemory/CryptoRF, useful where knowledge of different secrets provides different capabilities on the client. Each of these secrets is 64 bits long.

The companion chip also stores an authentication secret that the system can use to identify itself to the companion chip via the SHA-1 Hash algorithm. This secret is 128 bits long. In addition, each companion chip is shipped from Atmel with a guaranteed unique serial number, which is 120 bits long. Data sent to the chip during programming is protected by an encryption algorithm also based on SHA-1. The key used for this communication is 128 bits long.

Conclusion

Most system designers should carefully consider how to protect the investment in brand reputation, engineering expertise / time and intellectual property embodied in their product. In many cases, incorporating a hardware security component such as CryptoMemory or CryptoRF can provide significant protection against all but the most sophisticated attacker.

This is especially true where a future revenue stream depends on the continued sale of consumable, replaceable or add-in components. Where these components connect to the host via electrical connections, the simple two wire interface of CryptoMemory provides an easy way to integrate world-class cloning protection. When environmental factors such as dirt, moisture, chemicals, etc. exist, then CryptoRF, which incorporates the same robust crypto and secure storage capabilities with an RFID interface, is ideal.

For designers with extensive cryptographic experience and the necessary secure computation and storage elements within their system, implementing the host side functionality is straightforward. Atmel can help these engineers with pre-compiled libraries for some of the components.

These solutions provide the greatest degree of flexibility and can accommodate complicated or distributed architectures. When properly implemented, they are very secure.

For most designers and most systems, time to market, overall engineering cost and system cost are key factors and in this very common situation the CryptoMemory companion chip can prove quite beneficial.

Editor's Notes about Atmel Corporation

Founded in 1984, Atmel Corporation is headquartered in San Jose, California with manufacturing facilities in North America and Europe. Atmel designs, manufactures and markets worldwide, advanced logic, mixed-signal, nonvolatile memory and RF semiconductors. Atmel is also a leading provider of system-level integration semiconductor solutions using CMOS, BiCMOS, SiGe, and high-voltage BCDMOS process technologies.

Further information can be obtained from www.cryptomemory.com.

Contact: Kerry Maletsky, Business Unit Director Crypto Products, 1150 E. Cheyenne Mountain Blvd., Colorado Springs, CO 80906, U.S.A. Tel: (719) 540-1000, e-mail: CM-RFID.Atmel@cso.Atmel.com