APPLICATION NOTE

Symmetric Session Encryption Key Exchange

ATSHA204A

Introduction

Many systems need to communicate sensitive information over open channels. Examples include traffic light controllers where firmware upgrades take place over the air, power distribution grids where consumption and/or billing information is transmitted over power lines, and telemetry where patient data is transmitted over mobile telephone networks. Secure communication of sensitive information requires the transmitting system to ascertain the identity of the recipient (authentication) and scramble the information from view by others (encryption). Cryptography allows both tasks to be accomplished. But while many systems are capable of performing the underlying cryptographic processes, they lack the ability to securely share the cryptographic keys needed to establish trusted communication channels.

An alternative method for key exchange such as ECDH (Elliptic Curve Diffie-Hellman) can be easily implemented using the Atmel® CryptoAuthentication™ ATECC508A crypto element device, and is described in a different application note. The methodology described here uses symmetric techniques and can be implemented in ATSHA204A devices, which may be more attractive for certain systems. Please note that other CryptoAuthentication crypto elements, namely the ATECC108A and ATECC508A, are supersets of the ATSHA204A and can also perform the methodology described in this document.

This document explains how crypto element devices establish a root of trust from which systems can generate session encryption keys at one end and recover them at the other without actually transmitting the keys.

1 Establishing Root of Trust

Root of trust is shared secret knowledge that can serve as the basis for trusted transactions. Trusted transactions include secure authentication and confidential communication of sensitive information. The shared secret knowledge is usually in the form of a root key from which secure transaction keys derive. The crypto device offers the ability to embed non-readable and non-modifiable root keys in a manner that establishes root of trust.

The Atmel CryptoAuthentication™ devices are tamper-resistant security devices. In addition to having a True Random Number Generator (TRNG) and hardware protection against physical and environmental tampering, the crypto devices implement in secure hardware logic the SHA-256 algorithm, the most robust cryptographic algorithm endorsed by experts to date. Of the many features, the crypto devices ability to securely store secrets, generate true random numbers, and perform SHA-256 cryptographic hashes enables systems to securely communicate sensitive information over open channels. It does so by generating session encryption keys in one system and recovering the same keys in another without actually transmitting the keys. This completely eliminates the keys from the realm of attacks in the communication channel.

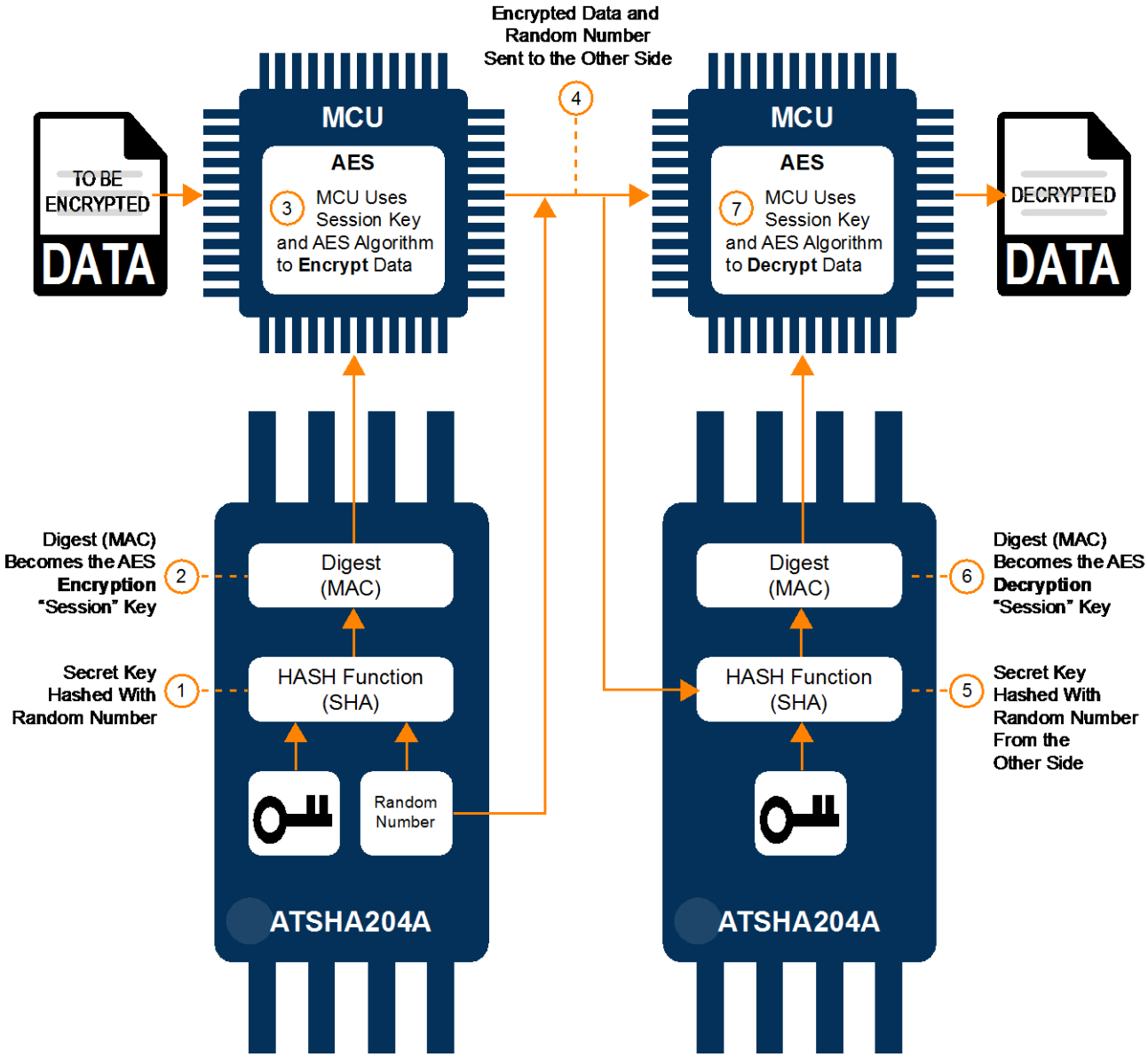
2 Session Encryption Key Generation, Recovery, and Secure Communication

All that is required to generate and recover session encryption keys between two or more securely communicating systems without actually transmitting said keys is a crypto device in each system. Through a process called device personalization, the system integrator gets to embed custom shared secrets within each crypto device and use hardware security mechanisms offered by the device to permanently seal the secrets from observation and/or modification. Each shared secret offers a unique root of trust among the systems.

Systems typically require only one root of trust, but having multiple ones can offer ability to differentiate between multiple services. For example, a system integrator may choose to use different root keys for communicating energy consumption and billing information over a power distribution network to accommodate the needs of respective servicing departments in a power distribution company.

Once the devices are in the systems, it takes a number of steps to accomplish secure end-to-end communication between two or more systems. The following sections explain these steps in greater detail, and the following figure illustrates how they all fit together.

Figure 2-1. Session Key Exchange



The ATSHA204A crypto element device can easily be used for secure session key exchange. The crypto element works together with an MCU that performs encryption/decryption with an algorithm such as AES. In order to create an encryption-decryption session, the MCU on each side will need an identical encryption-decryption key. To increase security, the key on each side should change with each session. That is why such keys are called session keys. The session key is securely exchanged from one side to the other. A random number is used to ensure that the keys will be different for each session.

Step 1 Secret Key Hashed with Random Number

The process starts by hashing the secret key stored in the ATSHA204A device with a random number created by the ATSHA204A. Note that the system may inject the random number into the crypto device, but Atmel advises using the crypto device's internal TRNG for high-quality random numbers to achieve the most effective session encryption keys. Effective session encryption keys are ones that virtually never repeat, thereby obviating replay and statistical attacks. This device will also emit the random number for later use. Generation of the session encryption key essentially opens a trusted communication session. The combination of the crypto device TRNG and the entropy properties of the SHA-256 cryptographic algorithm guarantee the quality of the session keys.

Step 2 Digest (MAC) Becomes AES Encryption "Session" Key

The result of the hashing operation will be a 32 byte Message Authentication Code (MAC). 16 bytes of that 32 byte (256 bit) MAC from ATSHA204A will be the AES session key is sent to the MCU.

Step 3 MCU Uses the Session Key and AES to Encrypt Data

The MCU runs an encryption algorithm such as the AES Algorithm over the data to be encrypted.

Step 4 Encrypted Data and Random Number Sent to the Other Side

The newly encrypted data and the random number are sent over to the other side so the data can be decrypted.

Step 5 Digest (MAC) Becomes the AES Decryption "Session" Key

To decrypt the message, the same key used to encrypt it must be used in the decryption process. That is exactly why the random number is sent over, which is to recreate the session key from that random number and the key stored on the ATSHA204A on the decrypting side. That is done by the random number being input to the SHA-256 hashing algorithm together with the key stored on the ATSHA204A. Because this is a symmetric operation, the secret keys stored on the ATSHA204A devices on both sides are identical. When the same random number is hashed with the same secret key, then the resulting 32-byte digest will be the same on the receiver (decrypting) side and on the sender (encrypting) side.

Step 6 Secret Key Hashed With Random Number From the Other Side

Just like the encrypting side, 16 bytes of the digest (i.e. MAC) represent the encryption/decryption key and is inputted into the MCU.

Step 7 MCU Uses Session Key and AES Algorithm to Decrypt Data

The receiving side's MCU runs the decryption algorithm (such as AES) using the session key to decrypt the encrypted code that was sent over from the other side. Disposition of the session encryption keys together with the earlier disposition of the random numbers by both the initiating and receiving systems essentially closes the trusted session. In other words, enough information no longer exists to ever recover or replay that session as long as the random number used was from a quality source like the crypto device's TRNG. By controlling how often a session key is generated in the initiating system, the system integrator effectively controls the session's lifetime determined by number of secure transactions, i.e., transmission and recovery of sensitive data.

3 End-to-End Authentication

The necessary requirements for securely communicating systems are the ability to authenticate recipients and encrypt sensitive information. The use of a crypto element device accomplishes both. While satisfaction of the encryption requirement is now evident, the same may not be apparent for authentication. Only an identically personalized crypto device can recover the session encryption key, and only authentic systems will be in possession of such personalized devices. Successful recovery of the session encryption key essentially proves the mutual authenticity of the initiating and target systems. A non-authentic system will not be able to recover the session encryption key, and, therefore, will not be able to gain access to the sensitive information. In other words, only mutually authentic systems can securely communicate, and authentication effectively derives from the shared root secret within the fortified confines of the crypto element device (i.e. in the secure key storage mechanisms).

4 Commands of Interest

The Atmel crypto element devices come with a rich set of commands that offer system integrators high flexibility in tackling various security challenges. After device personalization, only three of these commands are necessary to accomplish the application described in this document. They are the `Random`, `MAC`, and `Nonce` command, which are described in the following table.

Table 4-1. CryptoAuthentication Commands of Interest

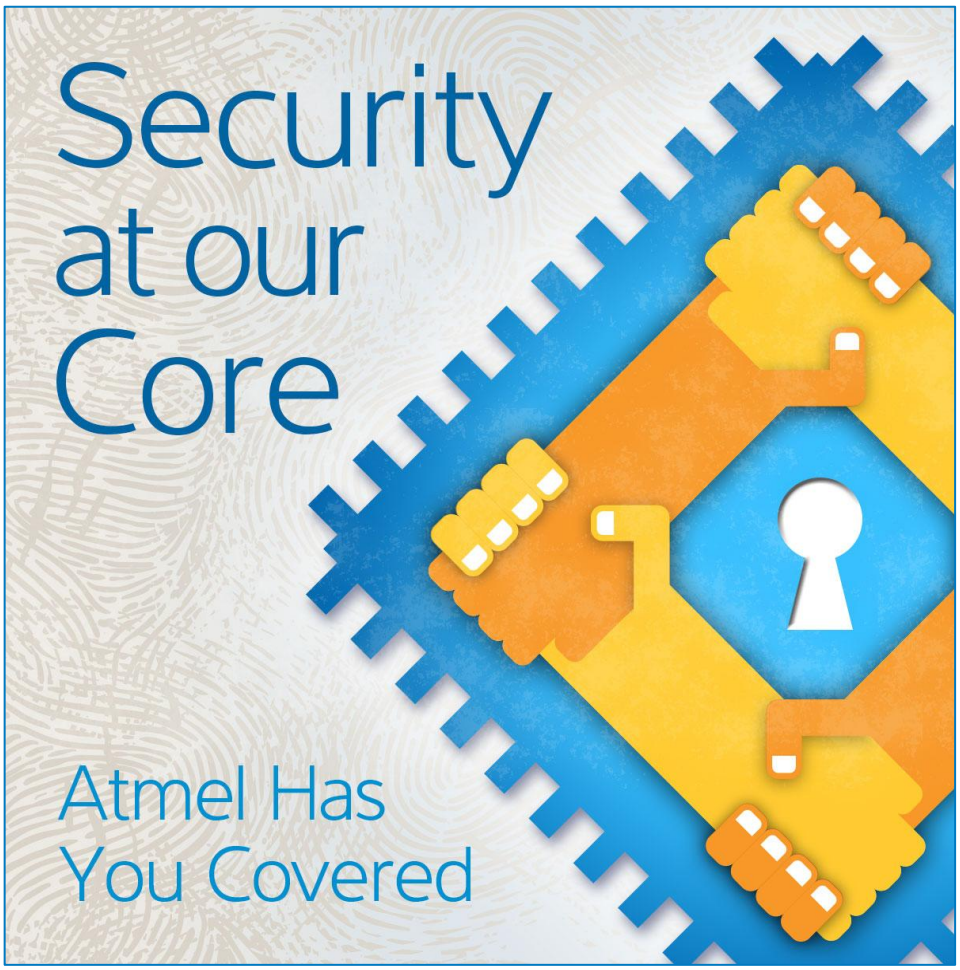
Command	Description
`Random`	Generates a random number for the creation of session encryption keys.
`MAC`	Combines the random number with the embedded root secret to create or recover the session encryption key.
`Nonce`	A precursor to the `MAC` command whose function is to initialize the crypto device into an internal state of high entropy. High entropy in this context is desirable because it eliminates the ease of guessing and mounting replay attacks. This miniscule list of commands emphasizes the ease of implementation and minimal demand on system resources like CPU bandwidth and code storage requirements.

4 Conclusion

Among its numerous features, the CryptoAuthentication product family offers a cost-effective approach for communicating systems to securely exchange session encryption keys over open channels without actually transmitting the keys themselves. It does so by establishing a root of trust between the communicating systems and enabling an initiating system to generate session encryption keys that are recoverable only by authentic receiving systems. The same device also establishes the mutual authenticity of the communicating systems. The devices offer ease of implementation with high-level commands, and demands very little in terms of system resources.

5 Revision History

Doc Rev.	Date	Comments
8777B	11/2015	Updated the application note, template, and CryptoAuthentication devices.
8777A	06/2011	Initial document release.



Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**