APPLICATION NOTE

Encrypted Reads and Writes

Atmel CryptoAuthentication

Introduction

The Atmel® CryptoAuthentication™ product line offers an exceptionally clean way of keeping traffic between the CryptoAuthentication device and microcontroller encrypted to prevent snooping on the bus during personalization or system operation. The encrypted read and encrypted write are configurations of the Read and Write commands and provide a mechanism for limiting access, enabling features, or updating a key value.

Features

- Securely Store Passwords or Keys without transferring the values in the clear
- Check Password or Keys without revealing expected value
1 Overview

Secure hardware devices can provide mechanisms to hide the clear value of the password, prevent offline
exhaustive attacks, and greatly increase the difficulty of local physical attacks. The Atmel®
CryptoAuthentication™ devices (crypto devices) provide such a capability in a very small package and at a low
cost which is easy to integrate into any digital system.

There are a couple of ways to implement the encrypted read or encrypted write commands.

2 Encrypted Read

2.1 Standard Encrypted Read

Figure 2-1. Standard Encrypted Read Flow Diagram

![Flow Diagram](image-url)
2.2 Simple Encrypted Read

Figure 2-2. Simple Encrypted Read Flow Diagram

`Nonce Command`  
Mode = Pass-through (0x03)  
Zero = 0x0000  
Numln = Nonce

`GenDig Command`  
Zone = Data (0x02)  
KeyID = Read Key ID

SHA-256  
Read Key, GenDig Opcode (0x15),  
Zone (0x02), Read Key ID, SN[8],  
SN[0:1], Zeros (25), Nonce, → Session Key

`Read Command`  
Zone = Data (0x02)  
Address = Slot and Block

XOR  
Cipher Text, Session Key, → Plain Text

SHA-256  
slot[KeyID] (Read Key), GenDig Opcode (0x15),  
Zone (0x02), KeyID (Read Key ID), SN[8],  
SN[0:1], Zeros (25), TempKey (Nonce)  
→ TempKey (Session Key)

XOR  
Plain Text (from Address),  
TempKey (Session Key) → Cipher Text

Numln → TempKey (Nonce)
3 Encrypted Write

3.1 Standard Encrypted Write

Figure 3-1. Standard Encrypted Write Flow Diagram
3.2 Simple Encrypted Writes

Figure 3-2. Simple Encrypted Writes Flow Diagram

- Encrypted Write
- Generate Nonce Input
- SHA-256 Write Key, GenDig Opcode (0x15), Zone (0x02), Write Key ID, SN[8], SN[0:1], Zeros (25), Nonce → Session Key
- XOR Plain Text, Session Key → Cipher Text
- SHA-256 Session Key, Write Opcode (0x12), Param1 (0x82), Param2 (Address), SN[8], SN[0:1], Zeros (25), Plain Text → Host MAC
- Write Command Zone = Data (0x82) Address = Slot and Block Value = Cipher Text MAC = Host MAC
- XOR Cipher Text, TempKey (Session Key) → Plain Text
- Error
- Success
- SHA-256 TempKey (Session Key), Write Opcode (0x12), Param1 (0x82), Param2 (Address), SN[8], SN[0:1], Zeros (25), Plain Text → Client MAC
- MAC (Host MAC) → Client MAC
- YES
- Write Plain Text to Address
- NO
- Write Opcode = Pass-through (0x03)
- Zero = 0x0000
- NumIn = Nonce
- SHA-256 slot(KeyID) (Write Key), GenDig Opcode (0x15), Zone (0x02), KeyID (Write Key ID), SN[8], SN[0:1], Zeros (25), TempKey (Nonce) → TempKey (Session Key)
4 Configuration

Before using the Atmel ATSHA204A device for encryption, there are initialization processes that are required to be performed. The initialization processes consist of personalizing and then locking the device. In the personalization step, the device behavior, the data slot behavior, and the data itself is configured as desired. After the personalization process is performed, the device is locked for the configuration to take effect and to prevent any further modification to the data. This section describes ways to configure the device for each specific encryption scheme.

4.1 Standard Encrypted Read and Simple Encrypted Read

Table 4-1. Standard Encrypted Read and Simple Encrypted Read

<table>
<thead>
<tr>
<th>Host</th>
<th>ATSHA204A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate Nonce Input (NumIn). 2. Save ReadKeyID and ReadKey.</td>
<td>1. Set:  ● SlotConfig.ReadKey (ReadKeyID)  ● SlotConfig.EncryptRead  ● SlotConfig.IsSecret</td>
</tr>
<tr>
<td></td>
<td>2. Lock Config Zone</td>
</tr>
<tr>
<td></td>
<td>3. Load ReadKey into Slot[ReadKeyID]</td>
</tr>
<tr>
<td></td>
<td>4. Lock Data Zone</td>
</tr>
</tbody>
</table>

Notes: 1. Encrypted Read only applies to 32-Bytes Read. 2. For Standard Encrypted Read, ReadKeyID can be either odd or even. If ReadKeyID is odd, CheckMacConfig bit corresponding to the Slot to Read must be zero. 3. For Simple Encrypted Read, ReadKeyID must be odd and CheckMacConfig bit corresponding to the Slot to Read must not be zero.

4.2 Standard Encrypted Write and Simple Encrypted Write

Table 4-2. Standard Encrypted Write and Simple Encrypted Write

<table>
<thead>
<tr>
<th>Host</th>
<th>ATSHA204A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate Nonce Input (NumIn). 2. Save WriteKeyID and WriteKey.</td>
<td>1. Set:  ● SlotConfig.WriteKey (WriteKeyID)  ● SlotConfig.IsSecret  ● Bit 14 of SlotConfig</td>
</tr>
<tr>
<td></td>
<td>2. Lock Config Zone</td>
</tr>
<tr>
<td></td>
<td>3. Load WriteKey into Slot[WriteKeyID]</td>
</tr>
<tr>
<td></td>
<td>4. Lock Data Zone</td>
</tr>
</tbody>
</table>

Notes: 1. Encrypted Write only applies to 32-Bytes Write. 2. If the Data Zone is unlocked, Param1 of Write Command is used to indicate whether or not the input data is encrypted. 3. For Standard Encrypted Write, WriteKeyID can be either odd or even. If WriteKeyID is odd, CheckMacConfig bit corresponding to the Slot to Write must be zero. 4. For Simple Encrypted Write, WriteKeyID must be odd and CheckMacConfig bit corresponding to the Slot to Write must not be zero.
## Revision History

<table>
<thead>
<tr>
<th>Doc Rev.</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>8981B</td>
<td>10/2015</td>
<td>Corrected the standard encrypted write flow diagram.</td>
</tr>
<tr>
<td>8981A</td>
<td>09/2015</td>
<td>Initial document release.</td>
</tr>
</tbody>
</table>