

CryptoAuthentication™ Firmware Protection



Atmel...Everywhere You Are...Securely

September 09

1



CRYPTOAUTHENTICATION

Agenda

- **Device basics**
 - Keys and OTP fuses
 - Operational block diagram
- **Firmware protection**
- **Categorizing Protection**
 - Protecting the question answer process
 - Firmware Updates
 - Network connected
 - Encryption and other runtime protection
 - Physical Protection



CRYPTOAUTHENTICATION

Device Basics

SA102S/10HS Secret Keys

Secret Keys

| | |
|--------|--|
| 0xFFFF | 0x 3B 4A FD 79 C4 C8 6C 78 25 A6 E9 AF D7 0F 2E D9 6E 3C 15 24 D3 EE 55 F6 D6 C8 86 F2 A0 2F B0 F6 |
| 0xA04D |,. |
| 0xE065 |,. |
| 0x13E7 |,. |
| |,. |
| |,. |
| |,. |

- Secret Keys hidden in Metal layers
- Never readable by any means
- Commands references Keys by index value
- Key provided to customer
 - Securely transmitted to customer by Atmel
 - One key provided to each customer
- Key 0xFFFF is published for testing,
 - The displayed value is correct for the first released SA102S Test parts

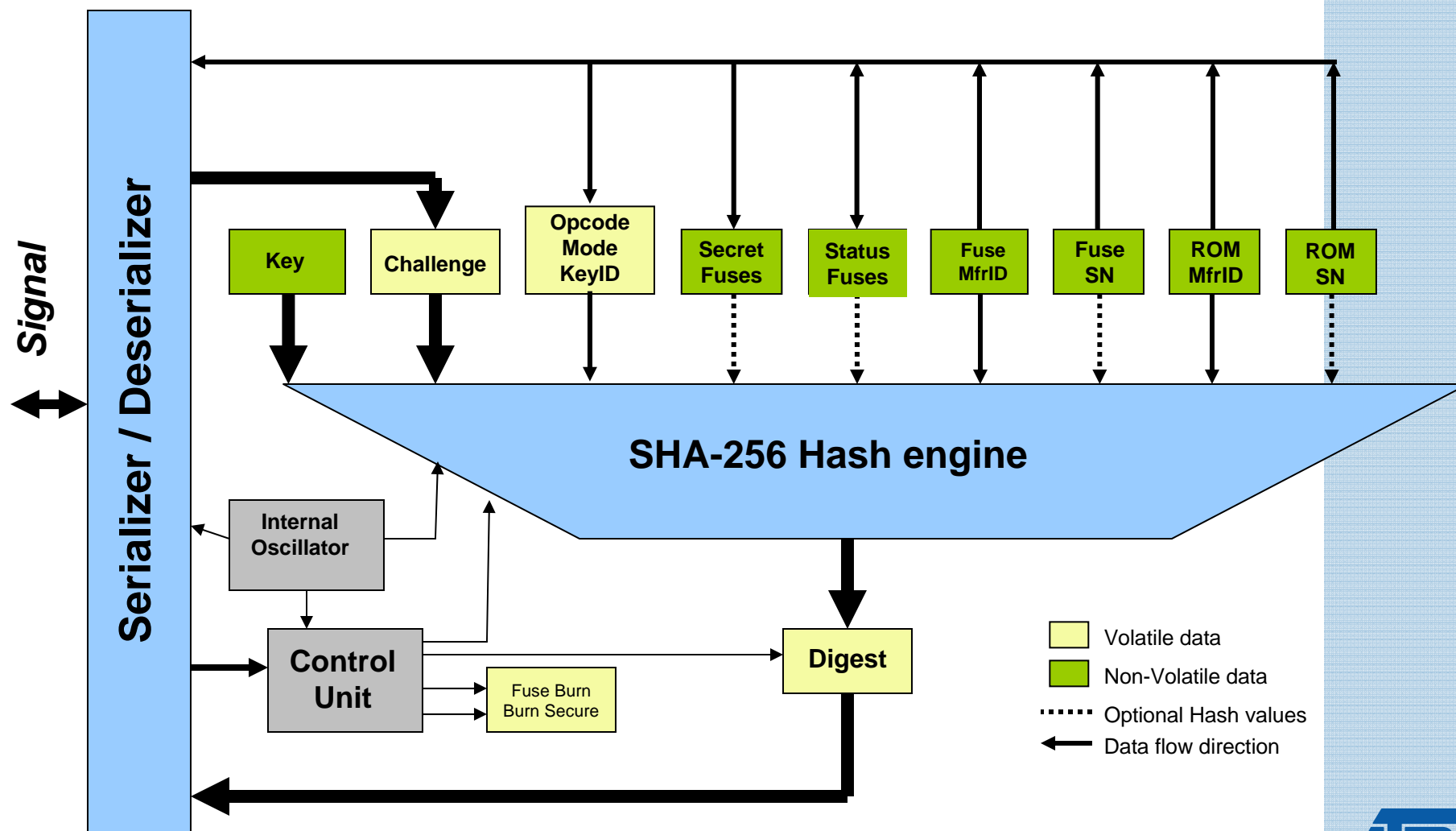
SA102S/10HS Fuse Mapping

SA102 Fuse Map

| | | | | | | | | | | | | | | | | | |
|-------------------|-----------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Read Address 0x00 | 0 – 15 | 0x00 0x0F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 16 – 31 | 0x10 0x1F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read Address 0x01 | 32 – 47 | 0x20 0x2F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 48 – 63 | 0x30 0x3F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read Address 0x02 | 64 – 79 | 0x40 0x4F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 80 – 95 | 0x50 0x5F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Read Address 0x03 | 96 – 111 | 0x60 0x6F | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| | 112 – 127 | 0x70 0x7F | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

| <i>Fuse #</i> | <i>Name</i> |
|---------------|--------------------|
| 1 | BurnFuse Disable |
| 0, 2 → 63 | Secret Fuses |
| 64 → 86 | Status Fuses |
| 87 | BurnSecure Disable |
| 88 → 95 | Fuse MfgID |
| 96 → 127 | Fuse SN |

SA102S/10HS Block Diagram



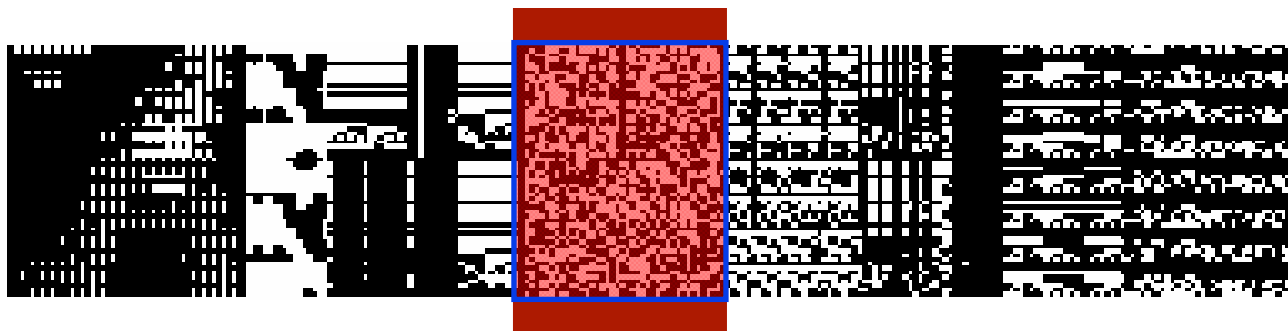
Why a hardware device

■ Add CryptoAuthentication based security

- Strong security with 256 bit key & SHA-256 standard algorithm
- Hardware-based authentication utilizes keys that **NEVER** leave the secure hardware crypto-logic and so are always hidden from attacks

■ Software is never a valid solution

- If it can be read it can be attacked



Detecting keys in software or on storage devices

Firmware Protection

- **Secure Microprocessors are the only 100% Solution**
- **Adequate level of security in 1995**
 - Not be a challenge to a entry level hacker
 - Security is a path not a destination
- **There is always a higher level of protection to be achieved**

Firmware Protection Solutions Approach

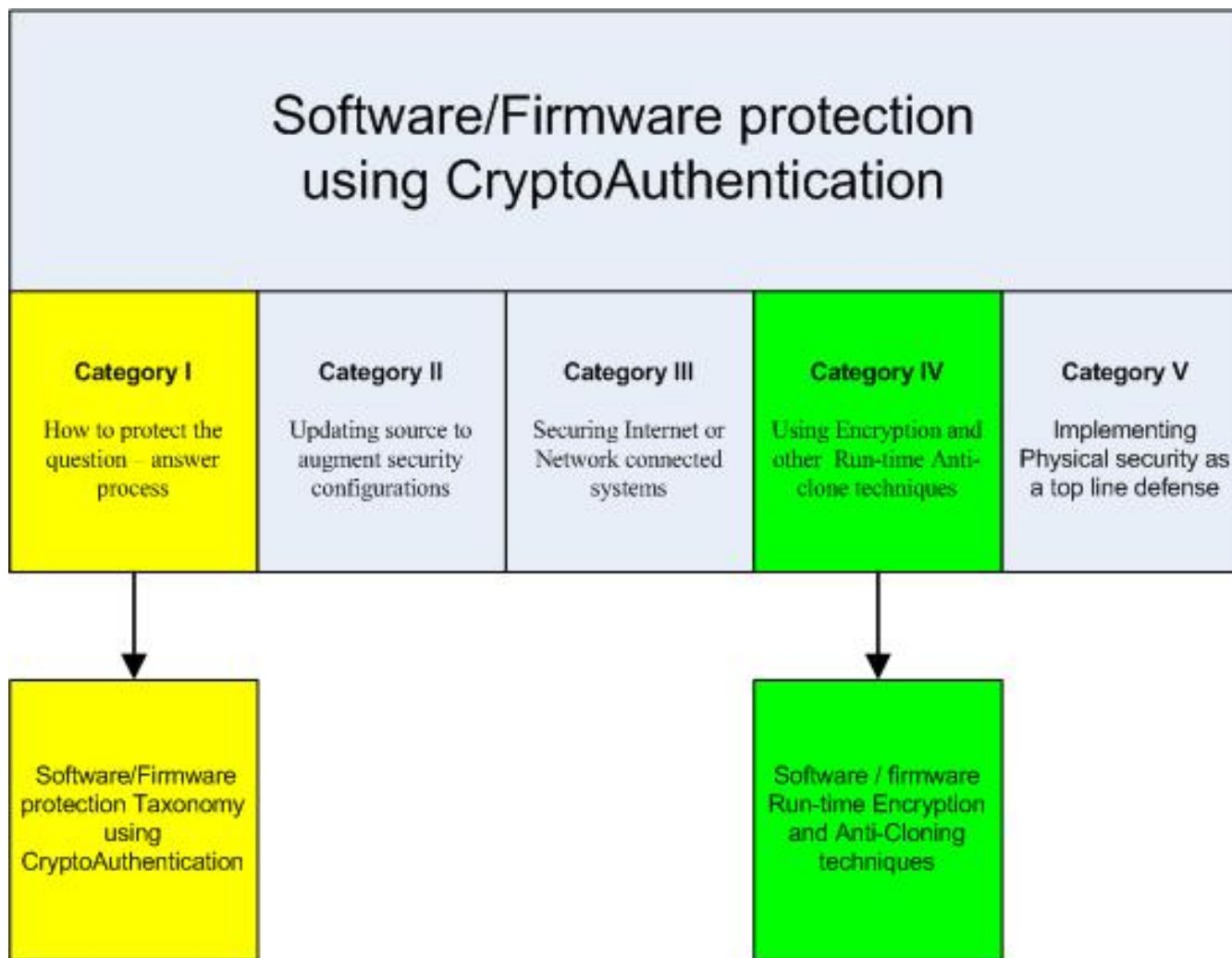
■ Provide an Implementation based on

- Utilizing one secure hardware device
- Provide an continuous structure to increase security

■ Provide security in Categories and Levels

- Best solution implements features of all categories
- Designers should always have the next level of security identified and ready to implement.

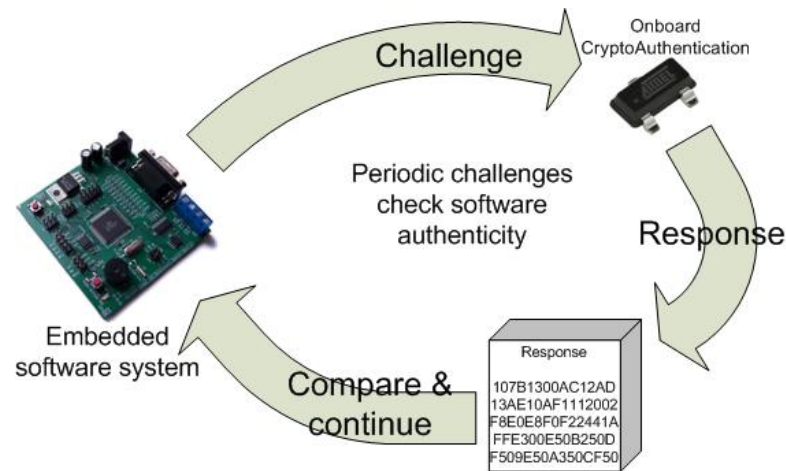
Categorizing Protection



Category I

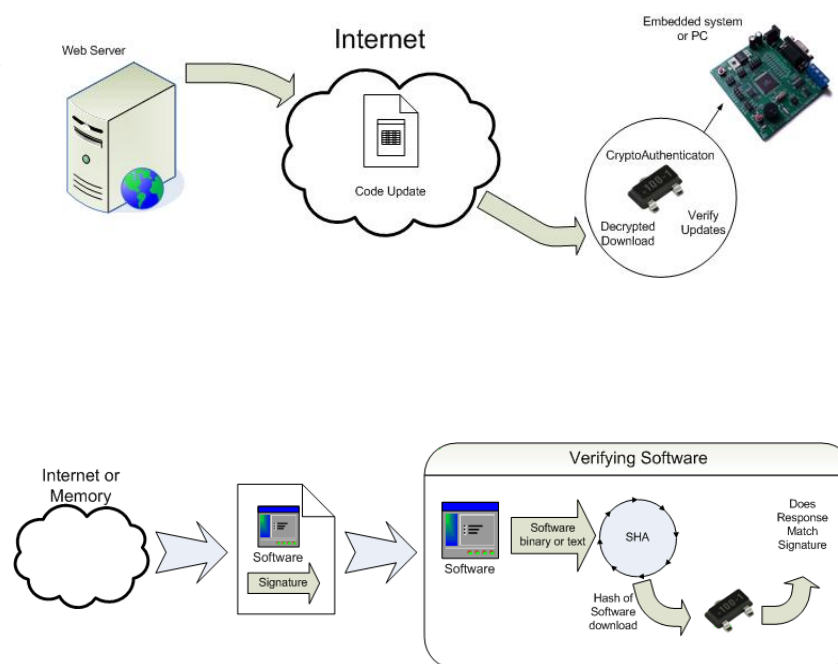
Protecting the question answer process

- **Level One – single challenge response to device**
- **Level Two – Make numerous challenge and responses**
- **Level Three – examine methods of obfuscating the calls.**
 - **Example: *firmware makes 10 calls to the security device use the 2nd response, XOR its value with the 5th response and then use that calculated value as the 9th challenge.***

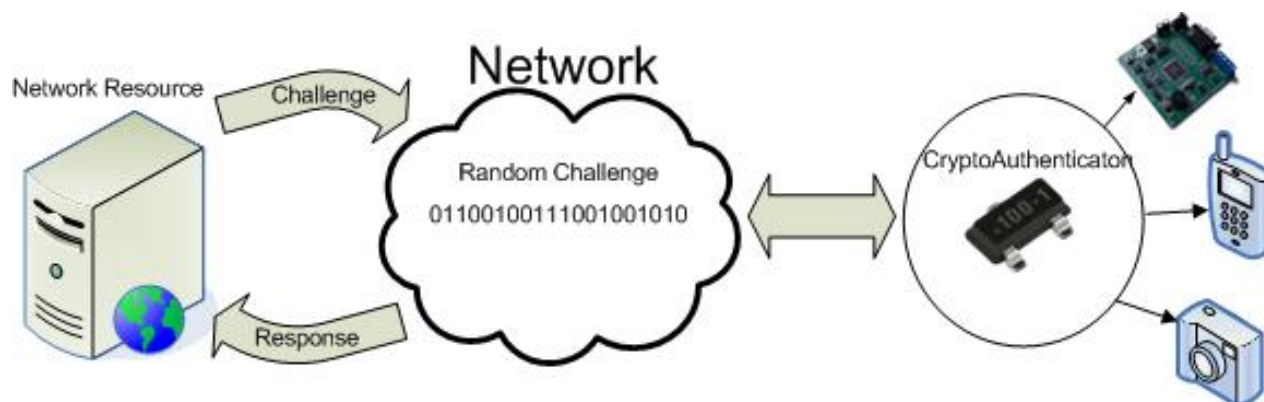


Category II Firmware Updates

- **Updating source to augment security configurations**
- **Level One** – Change the single challenge and response with each upgrade
- **Level Two** – Change the locations of multiple challenge-response calls within the firmware or change challenge response strategies
- **Advanced levels**
 - use runtime components
 - Add Verification of download
 - Encrypt update
- **Sends all current cloned devices back to ground zero**



Category III Network Connected

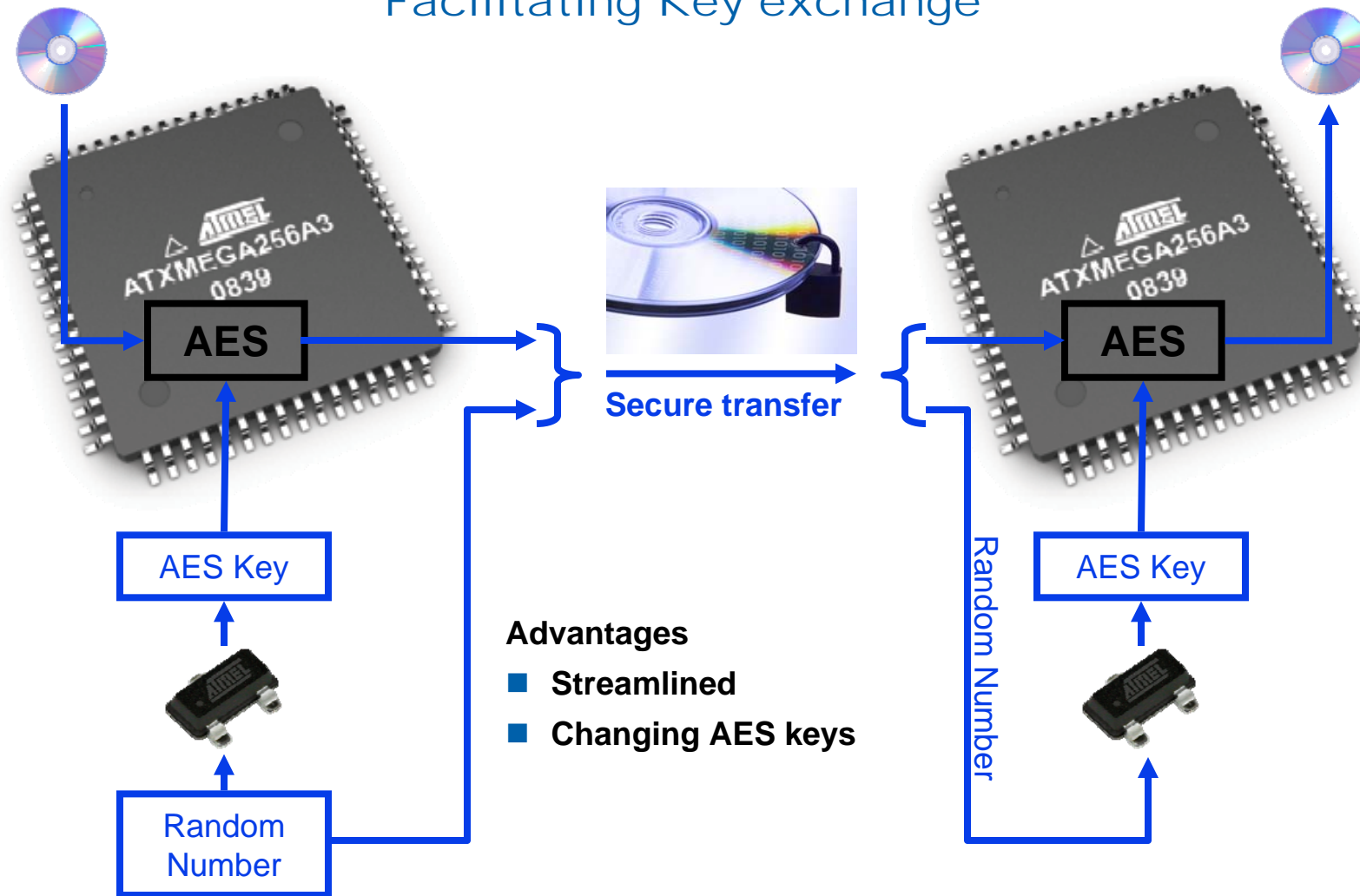


- **Per use direct connection to remote trusted system**
- **Level One** – Maintain a list of challenge-response pairs that change when connected to network, docking station, charger, PC, etc.
- **Level Two** – send verify message over network directly to remote system.
- **Advanced levels**
 - Remote resource logon
 - Help desk call in application
 - Encrypt content from remote system for individual product
- **Blacklist/Whitelist accessories , or Identify user tamper attempts.**

Category IV

Encryption and other runtime protection

Facilitating Key exchange





Category IV

Encryption and other runtime protection

- **Encrypted blocks of code that are decrypted at runtime prior to executing**
 - **Level one – encrypt some or all of the firmware and decrypt at runtime**
 - **Level two – Encrypt multiple blocks of code each with different random challenges**
 - **Advanced levels – Obfuscate Encryption schema**
 - Many blocks throughout the code
 - Blocks within blocks
 - Some blocks in rarely executed code
 - Decoy or non used blocks

Physical protection

- **Physical security of Host device**
- **CryptoAuthentication has no markings on package or Die**
- **Materials that protect or obfuscate physical board attacks**
- **Remove all device markings**
- **Control user access to systems**
 - **Physical tampers**
 - **High cost devices contain root of security**



Additional uses

■ Once the CryptoAuthentication device is installed it can be used later for adding security features

- Enable call center to remotely authenticate user prior to rendering service.
- Authenticate hardware or daughter cards
- Authenticating Battery packs
- Remote network logon or individual user authentication
- Encrypt confidential files produced by embedded systems
- Authenticate wireless nodes or encrypt communication between wireless nodes
- Authenticating nodes for devices communicating over power lines



Questions?