Security with STM32 & Secure Elements

Marco Sanfilippo
STMicroelectronics
EMEA MMS – Italy
Microcontroller Products
Technical Marketing
STM32 MCUs – Security what for ?

Main objectives
- Data confidentiality
- Data integrity
- IP protection
- Trusted execution

Protection from Software attacks
- Firmware corruption (Buffer overflows, stack corruptions …)
- Trojan horse, malware injection
- Untrusted firmware update
- Debug activation
- Wrong execution, denial of service

Protection from Hardware attacks

Non-Invasive :
- Fault injection by physical attacks on the system, external to package.
  - Out of range usage
  - Glitch on supplies or clocks
  - Radiation exposure
- Side channel attacks: spy product to get secrets (power supply, electromagnetic radiations,…)

Invasive :
- Internal fault injection after decapsulation (Force nodes by probing, laser beam, …)
- Reverse engineering (code/data extraction)
- Circuit modification (fib,…)

7 March 2016
STM32 MCUs – Security features 0/5

- CSS - Clock Security System
- ECC - Error Correction Code
- AT – Anti Tamper

and more….
## Today STM32 Security features 1/5

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefit</th>
<th>STM32 Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC calculation unit</td>
<td>Used to verify data transmission or storage integrity. Computes a signature of the software during runtime.</td>
<td>L0, L1, L4</td>
</tr>
<tr>
<td>Power Supply integrity monitoring</td>
<td>Ultra safe supply monitoring. POR/PDR/BOR/PVD Flag status to determine what causes reset (SW, watchdog, power up, low power, option bytes, ...)</td>
<td>F0,F1,F2,F3,F4,F7, L0,L1,L4</td>
</tr>
<tr>
<td>Read While Write</td>
<td>For efficient tamper detection logging</td>
<td>F1*,F4*,L0,L1*,L4</td>
</tr>
<tr>
<td>Clock Security System (CSS)</td>
<td>Independent clock sources and Clock recovery systems</td>
<td>F0,F1,F2,F3,F4,F7, L0,L1,L4</td>
</tr>
</tbody>
</table>
|                                 | CSS : Clock Security System
<p>|                                 | Internal clock available for secured program execution independently from external source clock (CSS) | F0,F1,F2,F3,F4,F7, L0,L1,L4 |
| Error Correction Code (ECC)     | Robust memory integrity. Hardened protection against fault injection attacks thanks to error detection | L0,L1,L4              |
| Parity check                    | Memory content integrity check. Hardened protection against fault injection attacks. | F0,F3,L4*             |
| Temperature Sensor              | Check if device is operating in expected temperature range. Hardened protection against temperature attacks. (AN3964). | F0,F1,F2,F3,F4,F7, L0,L1,L4 |
| Watchdogs                       | Independent watchdog and window watchdog for software timing control. Key registers to control watchdogs. | F0,F1,F2,F3,F4,F7, L0,L1,L4 |</p>
<table>
<thead>
<tr>
<th>Features</th>
<th>Benefit</th>
<th>STM32 Family</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Crypto</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Number Generator (RNG)</td>
<td>On chip entropy generation. Ensure strong keys, protect against replay attacks.</td>
<td>Based on DRBG-AES-128 F0,F1,F2,F3,F4,F7,L0,L1,L4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F2,F4,L0,L4,F7</td>
</tr>
<tr>
<td>Hashing Functions &amp; HMAC</td>
<td>Hash algorithm provides a way to guarantee the integrity of information, verify digital signatures and message authentication codes. MD5, SHA-1, SHA-224, SHA-256 (UM0586)</td>
<td>F0,F1,F2,F3,F4,F7,L0,L1,L4</td>
</tr>
<tr>
<td></td>
<td>. MD5, SHA-1, SHA-2</td>
<td>F2,F4*,F7</td>
</tr>
<tr>
<td>Symmetric Cryptography</td>
<td>AES-128 Bits (ECB, CBC,CTR)</td>
<td>F2,F4,F7,F7,L0,L1</td>
</tr>
<tr>
<td></td>
<td>AES- 128/256 Bits (ECB, CBC, CTR, GCM, GMAC, CMAC)</td>
<td>L4</td>
</tr>
<tr>
<td></td>
<td>STM32 cryptographic library package : (UM0586)</td>
<td>F0,F1,F2,F3,F4,F7,L0,L1,L4</td>
</tr>
<tr>
<td></td>
<td>▪ DES/TDES: ECB, CBC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ AES: ECB, CBC, CTR, CCM, CBC-MAC, GCM, CMAC, KEY WRAP</td>
<td></td>
</tr>
<tr>
<td>Asymmetric Cryptography</td>
<td>▪ RSA signature function with PKCS#1v1.5</td>
<td>F0,F1,F2,F3,F4,F7,L0,L1,L4</td>
</tr>
<tr>
<td></td>
<td>▪ ECC (Elliptic Curve Cryptography) : Key generation, Scalar multiplication, ECDSA. (UM0586)</td>
<td></td>
</tr>
</tbody>
</table>
## Today STM32 Security features 3/5

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefit</th>
<th>STM32 Family</th>
</tr>
</thead>
</table>
| Debug Lock Level 0,1,2 | **JTAG or SWD**  
Prevent unauthorized access to the device through debug interfaces.  
Highest security level is irreversible. ([AN4246](#)) | F0,F1*,F2,F3,F4,F7, L0,L1,L4 |
<p>| Anti Tamper | <strong>Protect against a wide range of physical attacks on HW system outside the MCU. (<a href="#">AN3371</a>)</strong> | F0,F1,F2,F3,F4,F7,L0,L1,L4 |
| Backup domain | <strong>Maintains tamper protection active even in Low Power modes. Multiple wake up sources. (<a href="#">AN3371</a>)</strong> | F0,F1,F2,F3,F4,F7,L0,L1,L4 |
| RTC (alarm timestamp) | <strong>Timestamp on tamper event. (<a href="#">AN3371</a>)</strong> | F0,F2,F3,F4,F7,L0,L1,L4 |
| RTC Register protection | <strong>Write protection. Unprotecting by writing a key sequence. Independent from system reset</strong> | F2,F3,F4,F7,L0,L1,L4 |
| Backup registers | <strong>For Confidential data storage (Keys …) Tamper automatically deletes registers content (<a href="#">AN3371</a>)</strong> | Backup register and SRAM See product datasheets |
| GPIO configuration locking | <strong>Lock of selected GPIO. Impossible to unlock until next reset. Capability to lock communication channels after tamper detection</strong> | F0,F1,F2,F3,F4,F7,L0,L1,L4 |</p>
<table>
<thead>
<tr>
<th>Features</th>
<th>Benefit</th>
<th>STM32 Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileges Permission Management</td>
<td>The processor MPU is a component for memory protection. It divides the memory map into a number of regions with privilege permissions and access rules.</td>
<td>F1*, F2, F3, F4, F7, L0, L1, L4</td>
</tr>
<tr>
<td>Firewall</td>
<td>Even more restrictive than MPU. Made to protect a specific part of code or data. Flash Memory, and/or to protect data into the SRAM from the rest of the code executed outside the protected area. <em>(AN4632)</em></td>
<td>L0, L4</td>
</tr>
<tr>
<td>Read Protection (RDP)</td>
<td>Global memory access control management. Prevents memory dumps, safeguarding user’s IPs. <em>(AN4246)</em></td>
<td>F0, F2, F3, F4, F7, L0, L1, L4, L4+SRAM</td>
</tr>
<tr>
<td>Write Protection (WRP)</td>
<td>Each sectors can be protected against unwanted write operations <em>(AN4246), AN4701(F4), AN4758(L4)</em></td>
<td>F0, F1, F2, F3, F4, F7, L0, L1, L4, L4+SRAM</td>
</tr>
<tr>
<td>Proprietary Code Protection (PCROP)</td>
<td>Each Sector can be configured in “execute only”. <em>(AN4246, AN4701(F4), AN4758(L4)</em></td>
<td>F4, L0, L1*, L4</td>
</tr>
<tr>
<td>Mass Erase</td>
<td>Safely remove IPs and confidential data. Force factory reset.</td>
<td>F7, L0, L1, L4</td>
</tr>
</tbody>
</table>
## Today STM32 Security features 5/5

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefit</th>
<th>STM32 Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traceability</td>
<td>Device electronic 96-bit Unique ID: Enables product traceability. Can be used for security key diversification.</td>
<td>F0,F1,F2,F3,F4,F7, L0,L1,L4</td>
</tr>
<tr>
<td>Secure Firmware Update</td>
<td>Software SFU: Secure firmware upgrade capability. (AN4023 &amp; AN4024)</td>
<td>F2,F4,L0,L4,F7</td>
</tr>
</tbody>
</table>
STM32 Crypto Library Package V3.1.0

MCD
Halim KACEM
Jasser MILED
Introduction

- The Crypto Library packages includes a set of cryptographic algorithms can run on all STM32 MCU series.

- STM32 Crypto library contains a software implementation of the cryptographic algorithms and also a hardware accelerators enhancement for some of them.

- STM32 cryptographic library files are provided in **object format** as a default delivery and **source code** under NDA license.
STM32 Crypto Library Approach (1/2)

- The STM32 crypto library V3.1.0 is divided in two category:
  - STM32 firmware crypto library V3.1.0
    - Based on **STM32 cube architecture**.
    - **All STM32 series** will be supported: STM32F0, STM32F1, STM32F2, STM32F3, STM32F4, STM32F7, STM32L0, STM32L1 and STM32L4.
    - All algorithms are based on **firmware implementation** without using any hardware acceleration.
    - The STM32 Firmware Crypto Library is distributed by ST as an **object code library**, accessed by the user application through an API.
    - The library is compiled for Cortex® **M0, M0+, M3, M4**, and **M7** cores.
    - The library is compiled with two optimization levels (**High size, High speed**).
    - Development Toolchains: **EWARM, MDK-ARM and GCC (Atollic)**.
  - STM32 hardware acceleration crypto library V3.1.0
    - Based on **STM32 cube architecture**.
    - Support all **STM32 series with hardware acceleration** : STM32F2, STM32F4, STM32F7, STM32L0, STM32L1 and STM32L4,
    - Support Only the algorithms based on **firmware implementation with hardware acceleration**.
    - The STM32 Hardware Acceleration Crypto library is distributed by ST as an **object code library**, accessed by the user application through an API.
STM32 Crypto Library Approach (2/2)

• The library is compiled for STM32 series F2, F4, F7, L0, L1, and L4.
• The library is compiled with two optimization levels (High size, High speed).
• Development Toolchains: EWARM, MDK-ARM and GCC Atollic/SW4STM32.

Algorithms fully supported by crypto hardware peripherals are not included in this package. To use them user can refer to dedicate STM32 Hal driver.
Crypto Hardware Peripheral supported in STM32 series

STM32 F series

June 2009
STM32F21x

AES(128-192-256) ECB, CBC, CTR
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1

RNG 32-bit
STM32F20x

STM32F415/417

AES(128-192-256) ECB, CBC, CTR
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1

RNG 32-bit
STM32F405/407

STM32F437/439

AES(128-192-256) ECB, CBC, CTR, CCM, GCM
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1, SHA-224/256

RNG 32-bit
STM32F427/429

STM32F745/756x

STM32L06x

STM32L05x

STM32L16x

STM32L14x

STM32L486x

STM32L471/476

STM32 L series

September 2011
STM32L1x

AES(128) ECB, CBC, CTR
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1

RNG 32-bit
STM32L2x

STM32L06x

February 2014
STM32L05x

AES(128) ECB, CBC, CTR
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1

RNG 32-bit
STM32L2x

STM32L486x

May 2015
STM32L471/476

AES(128-256) ECB, CBC, CTR, CMAC, GCM, GMAC
DES/TDES (64-128-192) ECB, CBC
Hash, HMAC MD5, SHA-1, SHA-224/256

RNG 32-bit
STM32 Crypto Library Package Structure

- STM32 Crypto Library Package V3.1.0:
STM32L4 Crypto Performances

- **Processing time**

![AES Processing Time (CPU Clock cycles)](chart)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pure Hardware Implementation</th>
<th>Pure Firmware implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-ECB consumption</td>
<td>13.5(mA) during 2.6µs</td>
<td>10.5(mA) during 34µs</td>
</tr>
</tbody>
</table>

**Consumption**

- Test conditions:
  - CPU = 80 MHz / IDE= IAR version 7.40 (High size).
  - Software based on Interrupt/CPU in sleep mode

Energy gain: x10 ratio SW/HW!
Libraries Standards/Quality

Compliant with standards

• NIST/FIPS standard
• ANSI-C source code
• ST coding
• Packaging rules
• ARM-CMSIS compliant for STM32

Crypto package quality

• CodeSonar tool analysis
Support
STM32 Crypto Package Download

• To Get the User manual and the software are available via this link

• Download the user manual UM1924

<table>
<thead>
<tr>
<th>User Manual</th>
<th>Version</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>UM1721: Developing Applications on STM32Cube with FatFs</td>
<td>2.2</td>
<td>516 KB</td>
</tr>
<tr>
<td>UM1722: Developing Applications on STM32Cube with RTOS</td>
<td>2.2</td>
<td>710 KB</td>
</tr>
<tr>
<td>UM1924: STM32 cryptographic firmware library</td>
<td>1.0</td>
<td>2385 KB</td>
</tr>
</tbody>
</table>

• Download the software

Get Software

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Version</th>
<th>Marketing Status</th>
<th>Order From ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-CUBE-CRYPTO LIB</td>
<td>3.0.0</td>
<td>Active</td>
<td>Download</td>
</tr>
</tbody>
</table>

(*) Suggested Resale Price per unit (USD) for BUDGETARY USE ONLY. For quotes, prices in local currency, please contact your local ST Sales Office or our Distributors
(**) The Material Declaration forms available on st.com may be generic documents based on the most commonly used package within a package family. For this reason, they may not be 100% accurate for a specific device. Please contact our sales support for information on specific devices.
Demonstration

Example Based on Firmware Library

• AES-128 CFB
• MDK-ARM v5.16
• STM32F103RB-Nucleo board

Example Based on Hardware Acceleration Library

• AES-128 CFB
• MDK-ARM v5.16
• STM324x9I-EVAL board with F439 device
ST Proposal

• Choose the robustness solution that matches the value of the secret to be protected

A secure element (SE) is typically a one chip secure microcontroller capable of securely hosting applications and their confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities.
Secure element – architecture proposal

IoT developer focuses on applications, the secure element handles the secrets
STSAFE-A: state of the art security

Fact base security evaluated by independent third parties – CC EAL5+
STSAFE-A: seamless integration

A comprehensive set of tools and services

- Secure MCU
- Secure O/S
- Personalization service
- Host library
- Example codes
Protecting against Attacks in MCU

- Identity theft
  - Tamper protection
  - Integrity
  - Traceability
- Deny of service
  - Throttling
- Data and Code spying
  - Memory protection
  - Privileges Permission Management
  - Debug levels
  - Tamper protection
  - Secure Firmware Update
- Data and Code modification
  - Memory protection
  - Debug levels
  - Tamper protection
  - Integrity
- Physical attach
  - Tamper protection
Thank you!