Customer Training Workshop
Traveo™ II Device Security
## Target Products

### Target product list for this training material

<table>
<thead>
<tr>
<th>Family Category</th>
<th>Series</th>
<th>Code Flash Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traveo™ II Automotive Body Controller Entry</td>
<td>CYT2B6</td>
<td>Up to 576KB</td>
</tr>
<tr>
<td>Traveo II Automotive Body Controller Entry</td>
<td>CYT2B7</td>
<td>Up to 1088KB</td>
</tr>
<tr>
<td>Traveo II Automotive Body Controller Entry</td>
<td>CYT2B9</td>
<td>Up to 2112KB</td>
</tr>
<tr>
<td>Traveo II Automotive Body Controller Entry</td>
<td>CYT2BL</td>
<td>Up to 4160KB</td>
</tr>
<tr>
<td>Traveo II Automotive Body Controller High</td>
<td>CYT3BB/4BB</td>
<td>Up to 4160KB</td>
</tr>
<tr>
<td>Traveo II Automotive Body Controller High</td>
<td>CYT4BF</td>
<td>Up to 8384KB</td>
</tr>
<tr>
<td>Traveo II Automotive Cluster</td>
<td>CYT3DL</td>
<td>Up to 4160KB</td>
</tr>
<tr>
<td>Traveo II Automotive Cluster</td>
<td>CYT4DN</td>
<td>Up to 6336KB</td>
</tr>
</tbody>
</table>
Introduction to Traveo II Body Controller Entry

- Device Security is in “CPU and Memory” and “Peripheral Interconnect”

**Hint Bar**

Review TRM chapters 13 and 27 for additional details
Introduction to Traveo II Body Controller High

Device Security is in “CPU and Memory” and “Peripheral Interconnect”

- CPU Subsystem:
  - Arm Cortex-M7: 350 MHz
  - eCT Flash: 8384 KB Code Flash + 256 KB Work Flash
  - SRAM0: 512 KB
  - SRAM1: 256 KB
  - SRAM2: 256 KB
  - ROM: 64 KB

- System Interconnect (Multi Layer AXI/AB, IPC, MPU/MO):
  - 115x TCPWM
  - 10x CANFD
  - 20x LIN
  - 1x FLEXRAY
  - eFuse

- Peripheral Interconnect (MMIO, PPU):
  - I2C, SPI, UART, LIN
  - I2S/TDM In/Out
  - Analog SAR ADC (12-bit)
  - SARMUX 96 ch

- System Resources:
  - Power
  - Sleep Control
  - POR BOD
  - REF
  - PWRSYS-HT
  - LDO
  - Clock
  - 2xPLL
  - IMO ECO
  - FLL
  - Test
  - Test Mode Entry
  - Digital DFT
  - Analog DFT
  - Power Modes
    - Active/Sleep
    - LowPower/Active/Sleep
    - DeepSleep
    - Hibernate

- High-Speed I/O Matrix, Smart I/O, Boundary Scan
  - 10x SPI
  - 10x CANFD
  - 10x LIN
  - 1x FLEXRAY
  - 2x ETH

- I/O Subsystem
  - Up to 196x GPIO_STD, 4x GPIO_ENH, 40xHSIO

- Hint Bar:
  - Review TRM chapters 13 and 27 for additional details
Introduction to Traveo II Cluster

Device Security is in “CPU and Memory” and “Peripheral Interconnect”

Review TRM chapters 13 and 27 for additional details
Device Security Overview

› Traveo™ II provides advanced security to protect user designs from unauthorized access and copying

› Features
  – Lifecycle stage
  – Memory and Peripheral Protection¹
    – Memory Protection Units (MPU)
    – Shared Memory Protection Unit (SMPU)
    – Peripheral Protection Units (PPU)
  – Flash Write and eFuse Read/Write Protection
    – Software Protection Units (SWPU)
  – Cryptography (Crypto) block

¹ For details of MPU, SMPU, and PPU see the Protection Unit section
Lifecycle Stages

- Traveo II has the following nonvolatile and irreversible lifecycle stages:
  - NORMAL_PROVISIONED
  - SECURE
  - SECURE_WITH_DEBUG
  - RMA
  - CORRUPTED
- ROM/Flash boot determines protection states based on the lifecycle stage
- DAP, PC1, and PCx are protected according to the protection state

Hint Bar
Review the Device Security TRM chapter for additional details
Lifecycle Stage Transitions

- Lifecycle stages transition by blowing the fuse of eFuse
- eFuse cannot be changed once programmed

Customer Delivery

NORMAL_PROVISIONED

SECURE_WITH_DEBUG

Blow D fuse by System Management API call

SECURE

Blow S fuse by System Management API call

RMA

Blow R fuse by System Management API call

CORRUPTED

Blow R fuse by System Management API call

Review the Device Security TRM chapter for additional details
## Lifecycle Stage Transitions

<table>
<thead>
<tr>
<th>Lifecycle Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL_PROVISIONED</td>
<td>Customers receive parts in this lifecycle stage.</td>
</tr>
<tr>
<td>SECURE</td>
<td>This is the lifecycle stage of a secure device. Access restrictions in SECURE mode are controlled by eFuse settings.</td>
</tr>
<tr>
<td>SECURE_WITH_DEBUG</td>
<td>This is similar to the SECURE lifecycle stage, except with NORMAL access restrictions applied to enable debugging, even if authentication fails. Devices that are in this stage are only used by developers and testers.</td>
</tr>
<tr>
<td>RMA</td>
<td>Devices can be brought into this stage so that Cypress can perform a failure analysis.</td>
</tr>
<tr>
<td>CORRUPTED</td>
<td>This stage is entered in case an error is detected when the boot process tries to determine the current lifecycle stage.</td>
</tr>
</tbody>
</table>
Memory and Peripheral Protection

Overview
- The MPU/SMPU/PPU restrict access to memory or peripheral address space
  - Internal attack protection
    - Prevents unauthorized code or bus masters from reading protection areas
  - External attack protection
    - Restricts access from unauthorized external equipment

Hint Bar
Review the Device Security TRM chapter and the Program and Debug Interface TRM chapter for additional details.
Use Case

- Allow debugger access to the CPU after password authentication
  - **Debugger sends the password to SRAM**
  - Generate a password authentication interrupt from the debugger
  - CM0+ executes password authentication
  - Allow access to the CPU after password authentication

Step 1: Send the password to SRAM

The debugger cannot access the CPU

Debugger

SWD/JTAG

CM4/7 Access Port

CM0+ Access Port

System Access Port

MPU

SRAM

CM0+

CM4/7

Traveo II
Memory and Peripheral Protection

Use Case

- Allow debugger access to the CPU after password authentication
  - Debugger sends the password to SRAM
  - Generate a password authentication interrupt from the debugger
  - CM0+ executes password authentication
  - Allow access to the CPU after password authentication
Memory and Peripheral Protection

Use Case

- Allow debugger access to the CPU after password authentication
  - Debugger sends the password to SRAM
  - Generate a password authentication interrupt from the debugger
  - **CM0+ executes password authentication**
  - Allow access to the CPU after password authentication

Step 1: Send the password to SRAM

Step 2: Generate a password authentication interrupt from the debugger

Step 3: Execute password authentication

The debugger cannot access the CPU
Memory and Peripheral Protection

Use Case

- Allow debugger access to the CPU after password authentication
  - Debugger sends the password to SRAM
  - Generate a password authentication interrupt from the debugger
  - CM0+ executes password authentication
- **Allow access to the CPU after password authentication**

(Note: The password authentication process is implemented by the user in software)
Flash Write and eFuse Read/Write Protection

› Overview
  - Traveo II has software protection units (SWPUs)\(^1\) that support:
    - Permissions for flash writing and erasing
    - Permissions for eFuse reading and writing

› Advantage
  - Prevents malicious or inadvertent modification of flash or eFuse
  - Prevents reading of eFuse protection data

\(^1\) SWPUs are stored in Supervisory Flash (SWPU details will be updated in a later revision)
Crypto Overview

- Crypto provides hardware implementation and acceleration of cryptographic functions
- Features
  - Cryptography function:
    - Symmetric key ciphers
    - Hashing
    - Asymmetric key ciphers
    - Pseudo-Random Number Generator (PRNG)
    - True Random Number Generator (TRNG)
    - Cyclic Redundancy Check (CRC)
  - Secure Hardware Extension (SHE)\(^1\)
  - Hardware Security Module (HSM)\(^1\)

\(^1\) The SHE and HSM solutions are provided by a third party
Crypto Block Diagram

Secure System Implementation (Call the Cryptography IP)

AHB-Lite Interface Logic
- Master Interface
- Slave Interface

AHB-Lite Interface

CM0+ → IPC → CM4/CM7

Interrupts

Instruction FIFO

Asymmetric
- RSA
- ECC

Symmetric
- DES
- TDES
- AES
- Chacha

Hashing
- SHA1
- SHA2
- SHA3

Others
- TRNG
- PRNG
- CRC

MMIO Registers

Hardware-Based Cryptography IP

Copyright © Infineon Technologies AG 2020. All rights reserved.
Hardware-Based Cryptography IP

› Asymmetric key ciphers
  – RSA and ECC

› Symmetric key ciphers
  – TDES: 64-bit length using a 64-bit key
  – AES: 128-bit length and programmable key length (128/192/256-bit key)
  – Chacha20: 512 random-looking bits
Hardware-Based Cryptography IP

- Hashing: SHA1, SHA2, and SHA3 hashes
- PRNG: Generate in a fixed range using three LFSRs
- TRNG: Generate using ring oscillators
- CRC: Programmable polynomial of up to 32 bits

**AHB-Lite Interface Logic**
- MMIO Registers
- Interrupts
  - Instruction FIFO
  - Asymmetric: RSA, ECC
  - Symmetric: DES, TDES, AES, Chacha
  - Hashing: SHA1, SHA2, SHA3
  - Others: TRNG, PRNG, CRC
Secure System Implementation

- The cryptography IP can be accessed only by the secure master (CM0+).
- Requests to CM0+ must be made via system calls using IPC from CM4/CM7.

**Diagram:**
- **CM0+** connected to **IPC** and **CM4/CM7** via AHB-Lite Interface.
- AHB-Lite Interface Logic includes:
  - Master Interface
  - Slave Interface
- **Crypto** includes:
  - Instruction FIFO
  - Asymmetric - RSA - ECC
  - Symmetric - DES - TDES - AES - Chacha
  - Hashing - SHA1 - SHA2 - SHA3
  - Hashing - SHA1 - SHA2 - SHA3
  - Others - TRNG - PRNG - CRC
- **Interrupts** to **MMIO Registers**
Digital Signature Verification with Asymmetric Key Ciphers

Use Case

- Traveo II can support digital signatures by controlling the hardware IP with software
  - Step 1: Distribute the public key

```
   (Server)
   (Private Key)  (Public Key)
   (Public Key)
   (Public Key)
   Traveo II

Step 1
```
Use Case

- Step 2:
  - First, using the hash function, calculate the hash value of the data.
  - Next, encrypt the hash value using the private key.
  - Finally, add the encrypted hash value as "signature" to the created data and send it.
Use Case

- Step 3:
  - First, decrypt the encrypted hash value using the public key.
  - Next, calculate the hash value of the received data using the same hash function as the sender.
  - Finally, compare the decrypted hash value with the calculated hash value. If they match, the data is correct.
Authentication of Communication Partner

Use Case
- Random numbers and encryption are used to authenticate the CAN communication partner
  - Step 1: Master ECU transmits the generated random number

---

**Step 1:** Transmit the random number to the Slave MCU

[Diagram showing the communication between Master MCU and Slave MCU with AES and RNG blocks, and multiple keys.]

**Hint Bar:**
Review the Cryptography Block TRM chapter for additional details
Authentication of Communication Partner

Use Case

- Step 2: The slave ECU encrypts the received random number with Key#3 and sends it to the master ECU

Master MCU

Crypto Block

RNG

AES

Keys

#1

#2

#3

#4

\ldots

#n

Slave MCU

Crypto Block

RNG

AES

Keys

#1

#2

#3

#4

\ldots

#n

Step 1: Transmit the random number to the Slave MCU

Step 2:
Encrypt the received random number with key #3 and send it to the Master MCU

Hint Bar

Review the Cryptography Block TRM chapter for additional details
Authentication of Communication Partner

Use Case

- Step 1: Transmit the random number to the Slave MCU
- Step 2: Encrypt the received random number with key #3 and send it to the Master MCU
- Step 3: The master ECU decrypts the received encrypted data with Key#3

Hints:
- Review the Cryptography Block TRM chapter for additional details.
Use Case
- Step 4: The master ECU checks whether the decrypted data and the transmitted RNG are the same, and authenticates the communication partner if they are the same.
Part of your life. Part of tomorrow.
# Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Submission Data</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>6152725</td>
<td>04/29/2018</td>
<td>Initial release</td>
</tr>
<tr>
<td>*A</td>
<td>6396994</td>
<td>07/31/2018</td>
<td>Added pages 2, 7, 8, 24, and 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated pages 3, 4, 5, 10, 11, 12, 13, 16, 17, 18, and 19</td>
</tr>
<tr>
<td>*B</td>
<td>6678028</td>
<td>09/18/2018</td>
<td>Added page 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated page 3, 4, 5, 20, 21, 22, 23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Delete CYT2B5 series</td>
</tr>
<tr>
<td>*C</td>
<td>6824363</td>
<td>03/04/2020</td>
<td>Added page: 20, 21, 22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated page: 3 to 26 (Hint Bar), 9 to 13 (Contents)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Deleted page: 27, 28 (delete the appendix pages)</td>
</tr>
<tr>
<td>*D</td>
<td>7082696</td>
<td>02/03/2021</td>
<td>Updated pages 2, 3, 4, 5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Convert content to IFX format</td>
</tr>
</tbody>
</table>