XFS4IoT SP-Dev Workgroup

4 Oct 2022

XFS4IoT SP-Dev Workgroup Agenda

- Recap from previous meeting
- End to End security: response security
- TPM framework – why and how
- What's next?

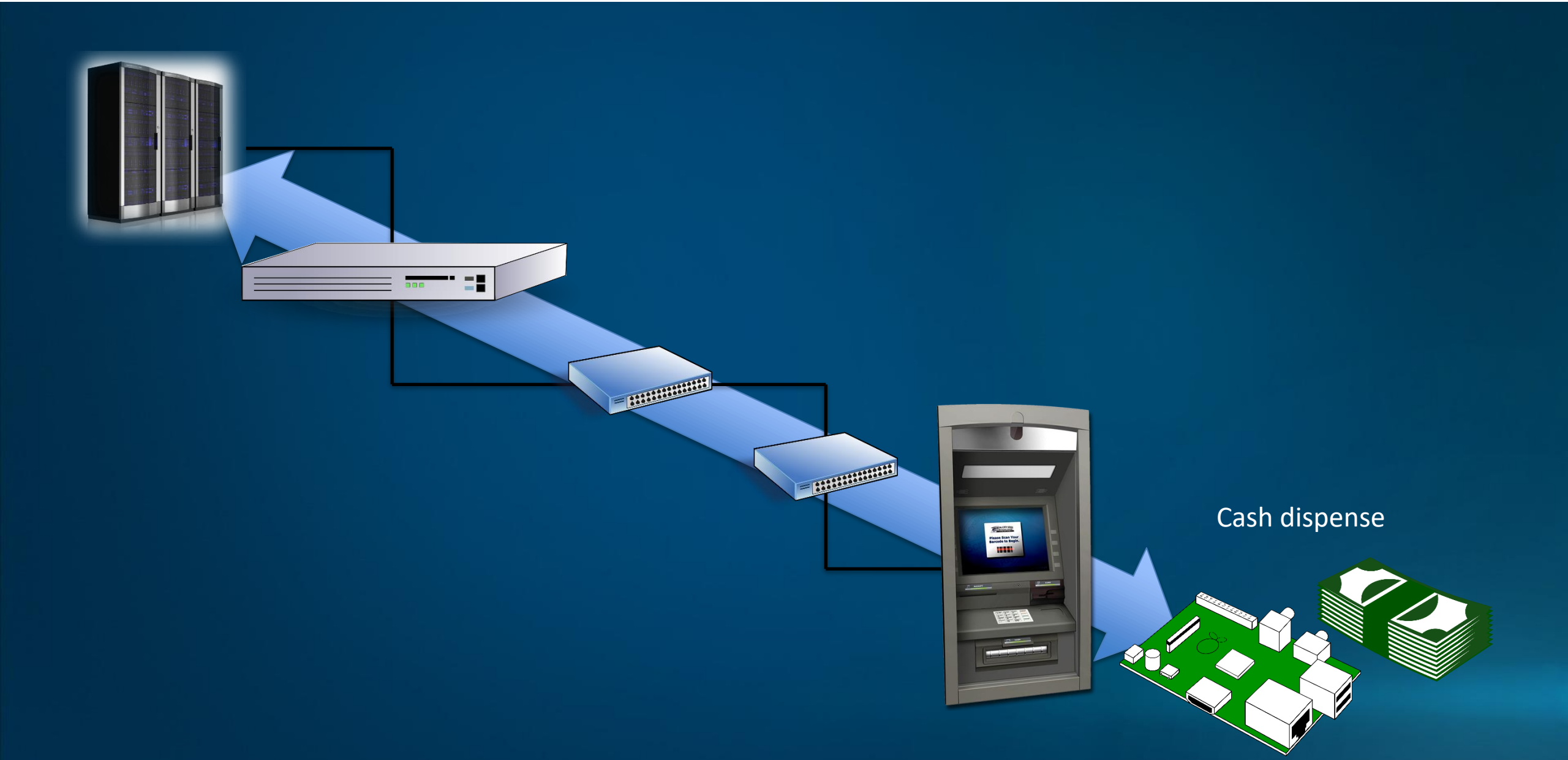
Recap from previous meeting

XFS4IoT Workgroup Review

- Provided update on the next CEN specification release
  - Originally planned for mid-2022, delayed to Q3 2022
    - E2E security (other device commands…)
    - Cheque deposit interface
    - Clarifications and update to Business plan
- Reviewed the latest changes in the SP-Dev Framework and Release 1.3.0
  - Numerous technical changes and fixes
  - Release 1.3.0 is now available via GitHub…and also published on nuget.org as usual

End to End security: response security

End to End security

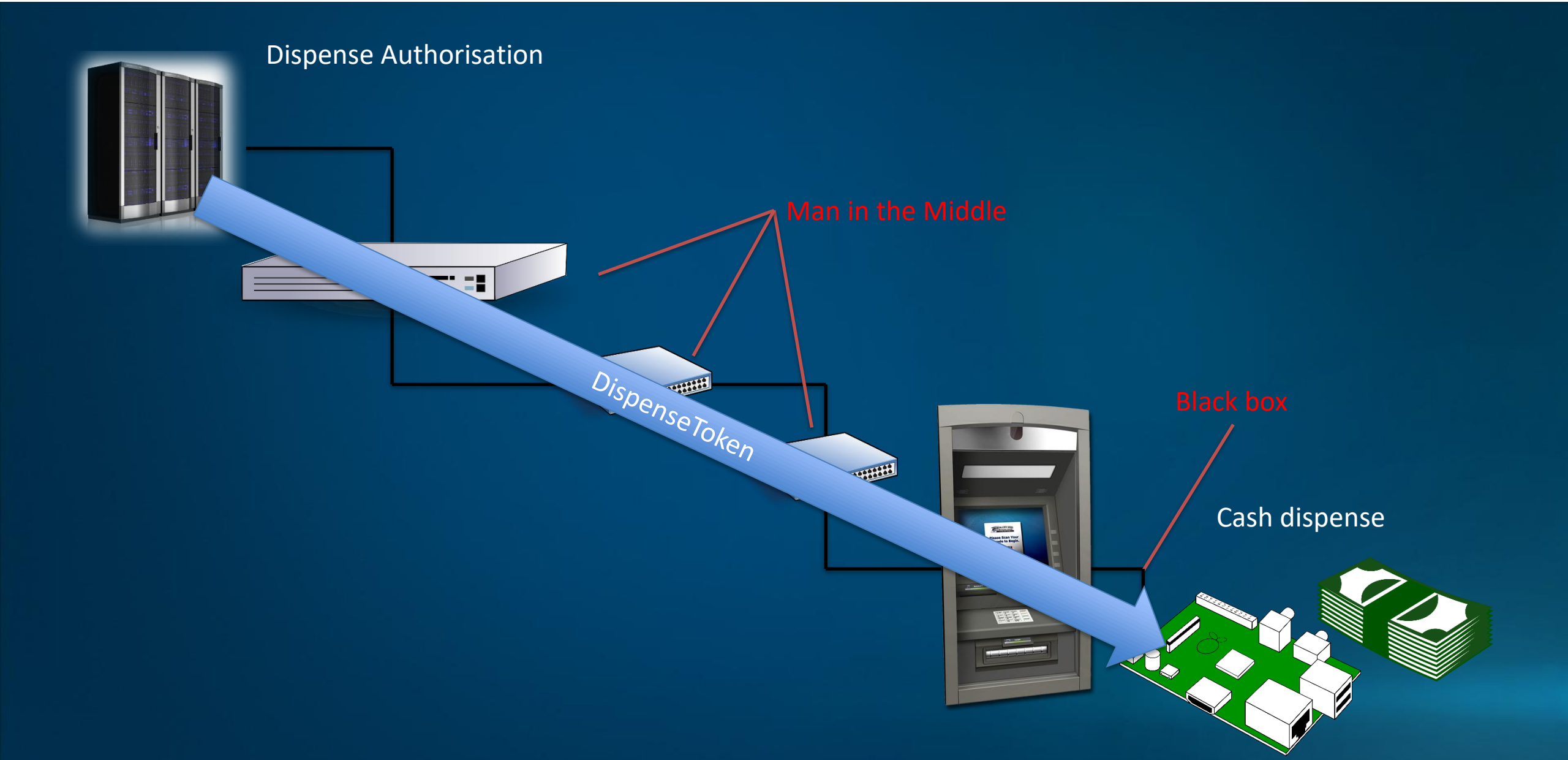


Known attacks

- Black box attacks
  - Between the PC and the dispenser
  - Attacker sends commands directly to the dispenser
- Network “Man in the Middle” attacks
  - Anywhere on the network outside the ATM
  - Send or change messages to make the ATM dispense cash

- Protected with a command token, “DispenseToken” attached to the dispense command

Command token

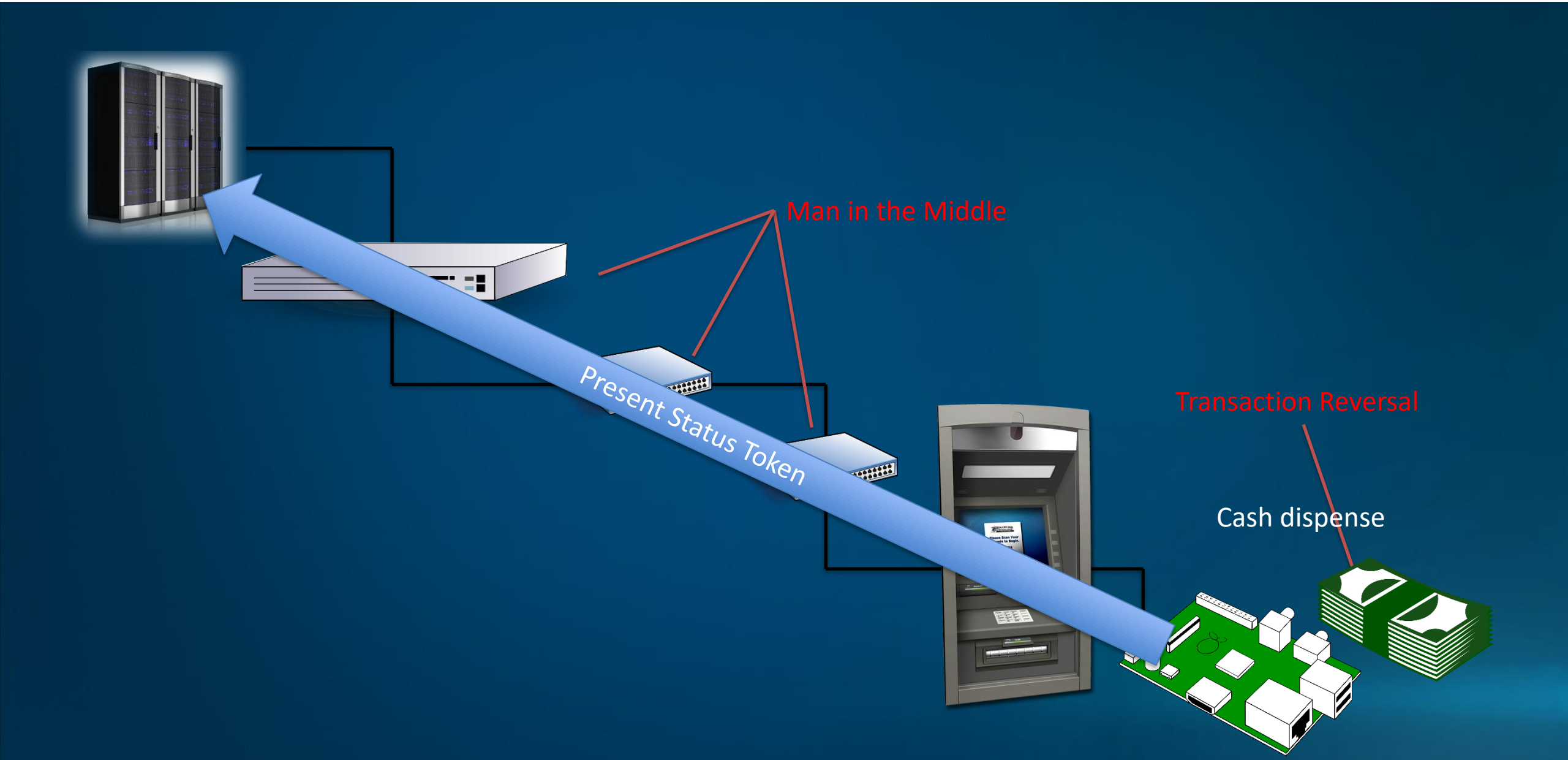


Potential attacks

"Transaction Reversal Fraud" + "Man in the Middle"

- **Transaction Reversal Fraud** – take cash, but reverse transaction so nothing is debited from the account
- **Man in the Middle** – Change messages on the network

Response token



Present status token

```
NONCE=1414,TOKENFORMAT=1,TOKENLENGTH=0268,DISPENSEID=CB735612FD6141213C2827FB5A
6A4F4846D7A7347B15434916FEA6AC16F3D2F2,DISPENSED1=50.00EUR,PRESENTED1=YES,PRESE
NTEDAMOUNT1=50.00EUR,RETRACTED1=YES,RETRACTEDAMOUNT1=?,HMACSHA256=55D123E9EE64F
0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83
```

```
NONCE=1414,
TOKENFORMAT=1,
TOKENLENGTH=0268,
DISPENSEID=CB735612FD6141213C2827FB5A6A4F4846D7A7347B15434916FEA6AC16F3D2F2,
DISPENSED1=50.00EUR,
PRESENTED1=YES,
PRESENTEDAMOUNT1=50.00EUR,
RETRACTED1=YES,
RETRACTEDAMOUNT1=?,
HMACSHA256=55D123E9EE64F0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83
```

Present status token

```
NONCE=1414,TOKENFORMAT=1,TOKENLENGTH=0268,DISPENSEID=CB735612FD6141213C2827FB5A
6A4F4846D7A7347B15434916FEA6AC16F3D2F2,DISPENSED1=50.00EUR,PRESENTED1=YES,PRESE
NTEDAMOUNT1=50.00EUR,RETRACTED1=YES,RETRACTEDAMOUNT1=?,HMACSHA256=55D123E9EE64F
0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83
```

NONCE=1414,
TOKENFORMAT=1,
TOKENLENGTH=0268,
DISPENSEID=CB735612FD6141213C2827FB5A6A4F4846D7A7347B15434916FEA6AC16F3D2F2,
DISPENSED1=50.00EUR,
PRESENTED1=YES,
PRESENTEDAMOUNT1=50.00EUR,
RETRACTED1=YES,
RETRACTEDAMOUNT1=?,
HMACSHA256=55D123E9EE64F0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83

Present status token

```
NONCE=1414,TOKENFORMAT=1,TOKENLENGTH=0268,DISPENSEID=CB735612FD6141213C2827FB5A
6A4F4846D7A7347B15434916FEA6AC16F3D2F2,DISPENSED1=50.00EUR,PRESENTED1=YES,PRESE
NTEDAMOUNT1=50.00EUR,RETRACTED1=YES,RETRACTEDAMOUNT1=?,HMACSHA256=55D123E9EE64F
0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83
```

```
NONCE=1414,
TOKENFORMAT=1,
TOKENLENGTH=0268,
DISPENSEID=CB735612FD6141213C2827FB5A6A4F4846D7A7347B15434916FEA6AC16F3D2F2,
DISPENSED1=50.00EUR,
PRESENTED1=YES,
PRESENTEDAMOUNT1=50.00EUR,
RETRACTED1=YES,
RETRACTEDAMOUNT1=?,
HMACSHA256=55D123E9EE64F0CC3D1CD4F953348B441E521BBACCD6998C6F51D645D71E6C83
```

SP-Dev framework support

- Currently being implemented
- Framework will support ‘Response’ tokens, including GetPresentStatus token
- Implementation will be part of the ‘end to end’ security package
- Written in C11, to run on low power hardware
- *Must* be part of the firmware – inside the safe

TPM framework – why and how

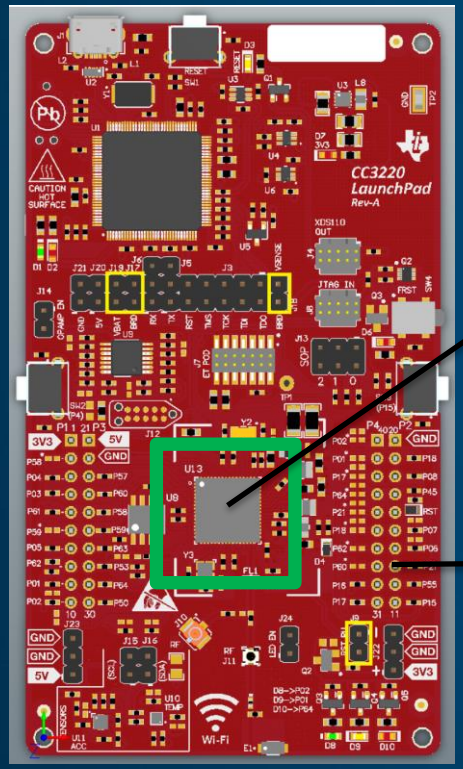
Why using a TPM?

- TPMs are the root of all security
- No TPM = no security!

Whitepaper available here : https://www.kal.com/en/tpm-whitepaper

E2E security on real hardware

Hardware components



Ti LaunchPad development board

SPI protocol

CC3220SF

- Arm® Cortex®-M4 core
- 256KB of RAM
- UART, I2S, I2C, SPI, SD, ADC
- Cost less than 5USD…

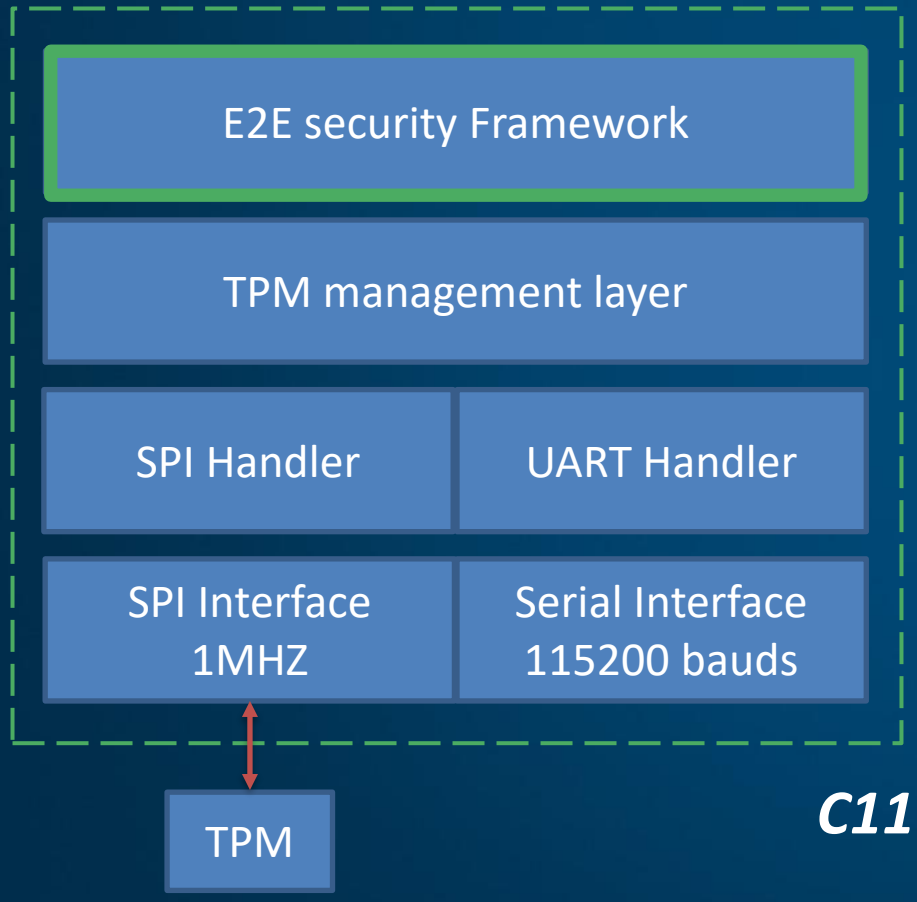
ATTPM20P

- Trusted Platform Module
- Version 2.0 of TCG
- (SPI) Protocol up to 36 MHz
- RNG, HMAC, AES, SHA-256, ECC, and RSA
- Cost less than 2USD…

References:
https://www.ti.com/launchpad
https://www.microchip.com/en-us/product/ATTPM20P

TPM – E2E Security framework

Integration layers



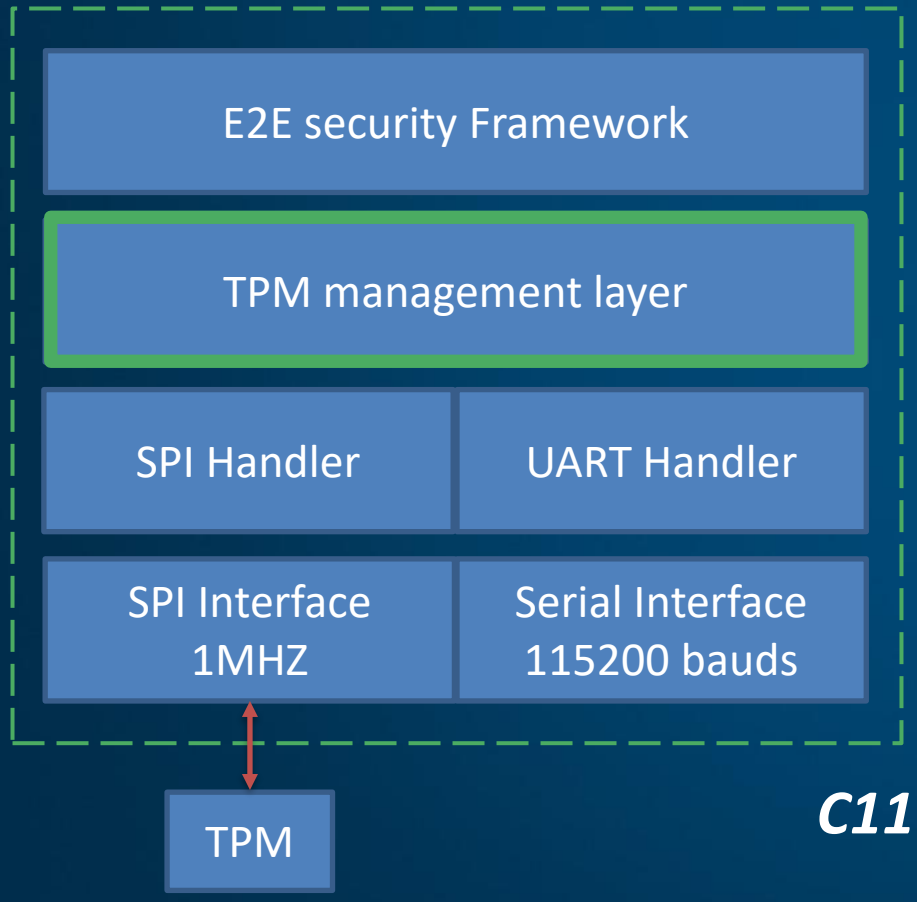
E2E Implemented Functions

CompareNonce	`extern bool CompareNonce(char const* const CommandNonce, size_t NonceLength)` `{`
FatalError	`extern void FatalError(char const* const Message)` `{`
Log	`extern void Log(char const* const Message)` `{`
ClearNonce	`extern void ClearNonce()` `{`
NewNonce	`extern void NewNonce( char const ** Nonce )` `{`
CheckHMAC	`extern bool CheckHMAC(char const *const Token, unsigned int TokenLength, char const *const TokenHMAC)` `{`

- ❑ Token verification done by E2E framework implementation
- ❑ E2E framework implementation inside MCU the entire SP framework is running in another device
- ❑ Dispense token validation (HMAC validation)

TPM – E2E Security framework

Integration layers



TPM management layer

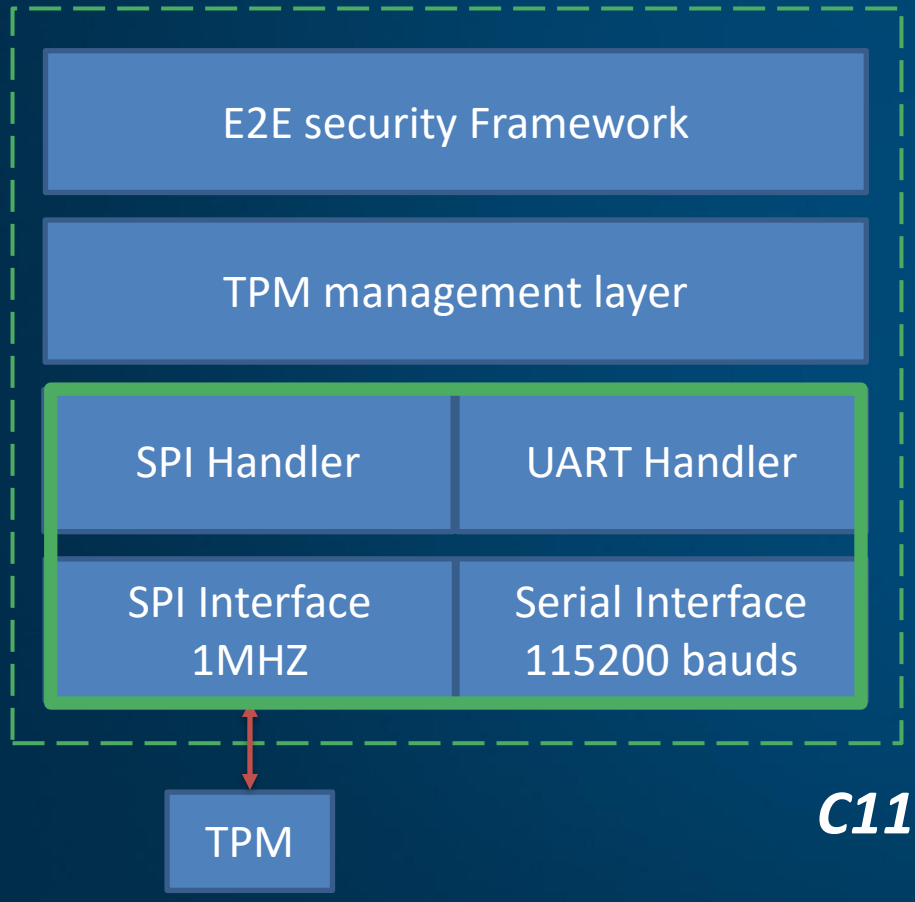
- FIFO interface implementation (SPI protocol)
- TPM commands
  - Startup sequence

  - Provisioning process (Known Symmetric key directly loaded in TPM)

  - Operation Mode

TPM – E2E Security framework

Integration layers



C11

SPI – UART interfaces

SPI Interface

```
SPI_Handle      masterSpi;

void InitSPI(void)
{
    SPI_Params      spiParams;

    // Init SPI Module
    SPI_init();

    /* Open SPI as master and configure SPI bus */
    SPI_Params_init(&spiParams);
    spiParams.mode = SPI_MASTER;
    spiParams.frameFormat = SPI_POL0_PHA0;
    spiParams.bitRate = 1000000;
    spiParams.dataSize = 8;

    // Open SPI handler
    masterSpi = SPI_open(CONFIG_SPI_0, &spiParams);
    if (masterSpi == NULL) {
        UART_PRINT("\n\rError initializing master SPI");
    }
    else
        UART_PRINT("\n\rMaster SPI initialized\n\r");
}
```

UART Interface

```
static UART_Handle uartHandle;

UART_Handle InitUart(void)
{
    UART_Params uartParams;

    UART_init();
    UART_Params_init(&uartParams);

    uartParams.writeDataMode = UART_DATA_BINARY;
    uartParams.readDataMode = UART_DATA_BINARY;
    uartParams.readReturnMode = UART_RETURN_FULL;
    uartParams.readEcho = UART_ECHO_OFF;
    uartParams.baudRate = 115200;
    uartParams.dataLength = 8;

    uartHandle = UART_open(CONFIG_UART_0, &uartParams);
    /* remove uart receive from LPDS dependency */
    UART_control(uartHandle, UART_CMD_RXDISABLE, NULL);

    return(uartHandle);
}
```

E2E security on real hardware

Demo Simulated Host

- Console Application
- Connected through UART (COM4 port)

Idle – Waiting serial Data

Compute Token
Wait Request

Compute BAD Token

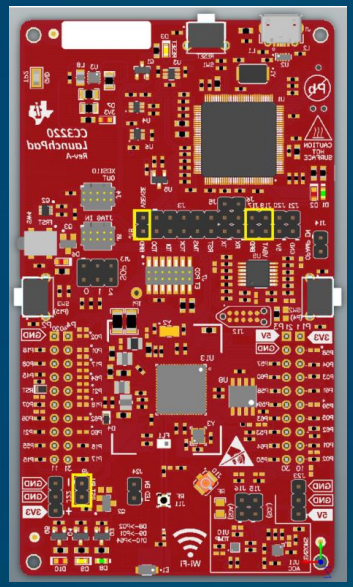
Send Random Nonce

Token Request 01

GOOD token response

Token Request 02

BAD token response



Start Initialization Sequence
Generates random nonce (TPM)

Initialization completed

Token Validation **OK**
Parse Token and Blink LED

Token Validation **FAILS**

HOST

DEVICE

E2E Security Framework implementation

Demo video is available on YouTube

YouTube channel:

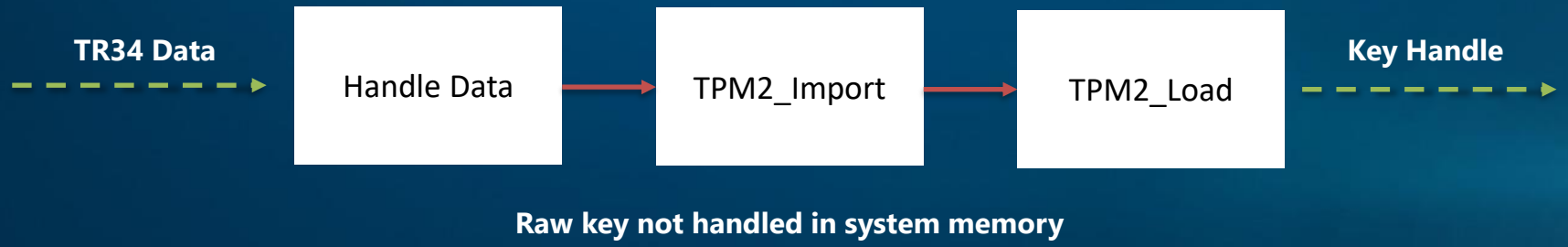
https://www.youtube.com/user/ATMsoftware/videos

TPM framework integration – TR34

Where we are?

- Extending functionality to support TR34

  — Hierarchy structure to store Asymmetric RSA key used to handle TR34 data
  — Implementing required TPM commands sequence to avoid handling raw key values in the system memory
  — Implementing TPM mechanism to import and load the received symmetric key



TPM into XFS4IoT SP framework

What's next?

- Full TR34 Demo implementation on real hardware.
- Include the TPM into XFS4IoT SP framework.
  - HW Target → TPM2.0. (ATTPM20T Microchip)
  - Implementation of FIFO interface through SPI protocol.
    - Using open-source library or implement from scratch.
  - Main TPM commands implementation (Defined in TCG specs)
    - Initialization commands, Provisioning commands, working commands.
  - Make the TPM layer available to be used as a Crypto-processor for any device class Cash Dispenser, Card Reader, Receipt Printer, etc.

What’s next?

Work in progress

- Working with the CEN XFS Committee to finalise the next specification release for Q3 2022
- Keep the SP-Dev framework up to date with CEN XFS specification
- Finalise E2E security support and progress on the TPM framework
- Continue to fine-tune SP-Dev framework through experience/feedback from XFS4 SPs development and workgroup members

Next call

Zoom

- First Tuesday of each month at 1300 UK time for 30 mins

Next call: 1st November 2022, 1300 UK, 0900 US EST, 2200 Tokyo time

Please note the change of time for US EST and Tokyo making it one hour later for both

Calls are 30 mins long

We will continue to use Zoom

(Interpretation in Japanese, Chinese and Spanish available using Zoom's interpretation feature)