# Mifare DESFire EV1 (in AES mode) with TRF7970AEVM and MSP430

**Texas Instruments**

**MSP430 & NFC/RFID Applications Teams**

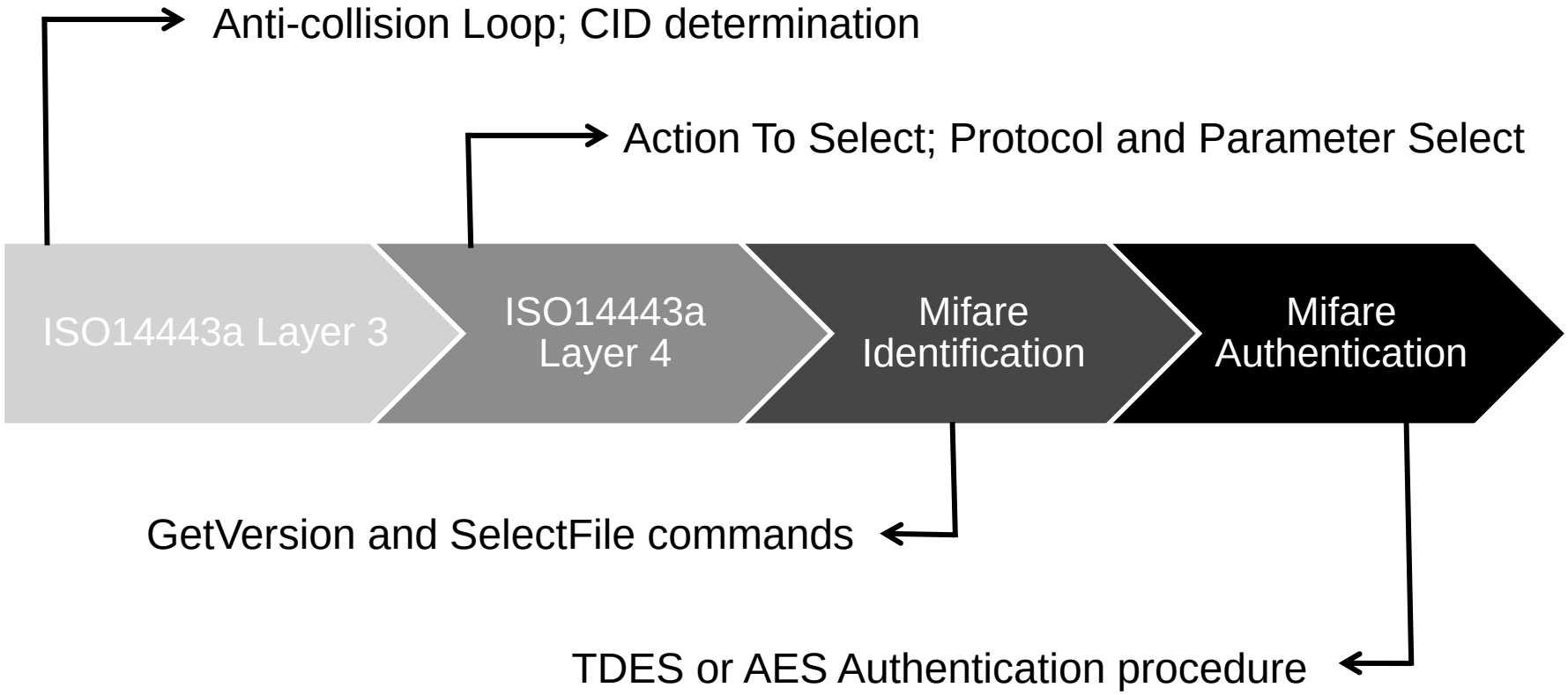
# Agenda

- Project Objective
- Protocol Description
- Project Evolution
- Customer Use
- Improvements

# Project Objective

- Objective:
  - Get working demo of DESFire cards being set into AES mode and authenticating on our hardware.
  - Target MCUs
    - MSP430G2553
      - for Authentication demo, low cost target for customers
    - MSP430F2370
      - used for setting card in AES mode due to recursion needed for MFDESFireEV1 DES algorithm, more RAM than value line
  - Target Transceivers
    - TRF7970A
    - TRF7964A

# Communication Structure

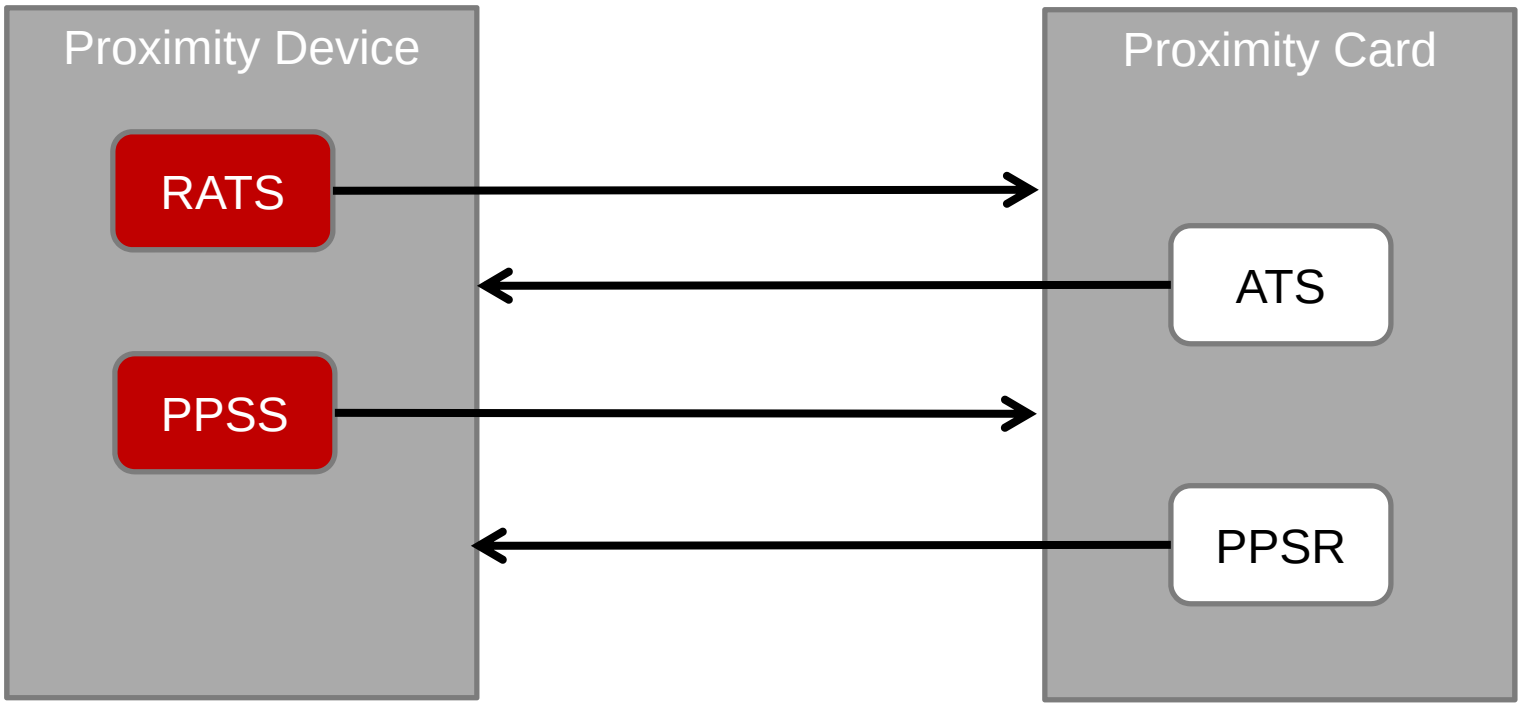


# Firmware Beginning State

- Started with TRF7970 Demo Code
  - Displays ISO14443A, ISO14443B, and NFC Protocols
  - ISO14443A only implemented to Layer 3
  - Layer 3 code not complete
    - CID not complete
    - No checks for completion
- Actions:
  - Trim code for ISO14443A only
  - Complete Layer 3 and add Layer 4 hook

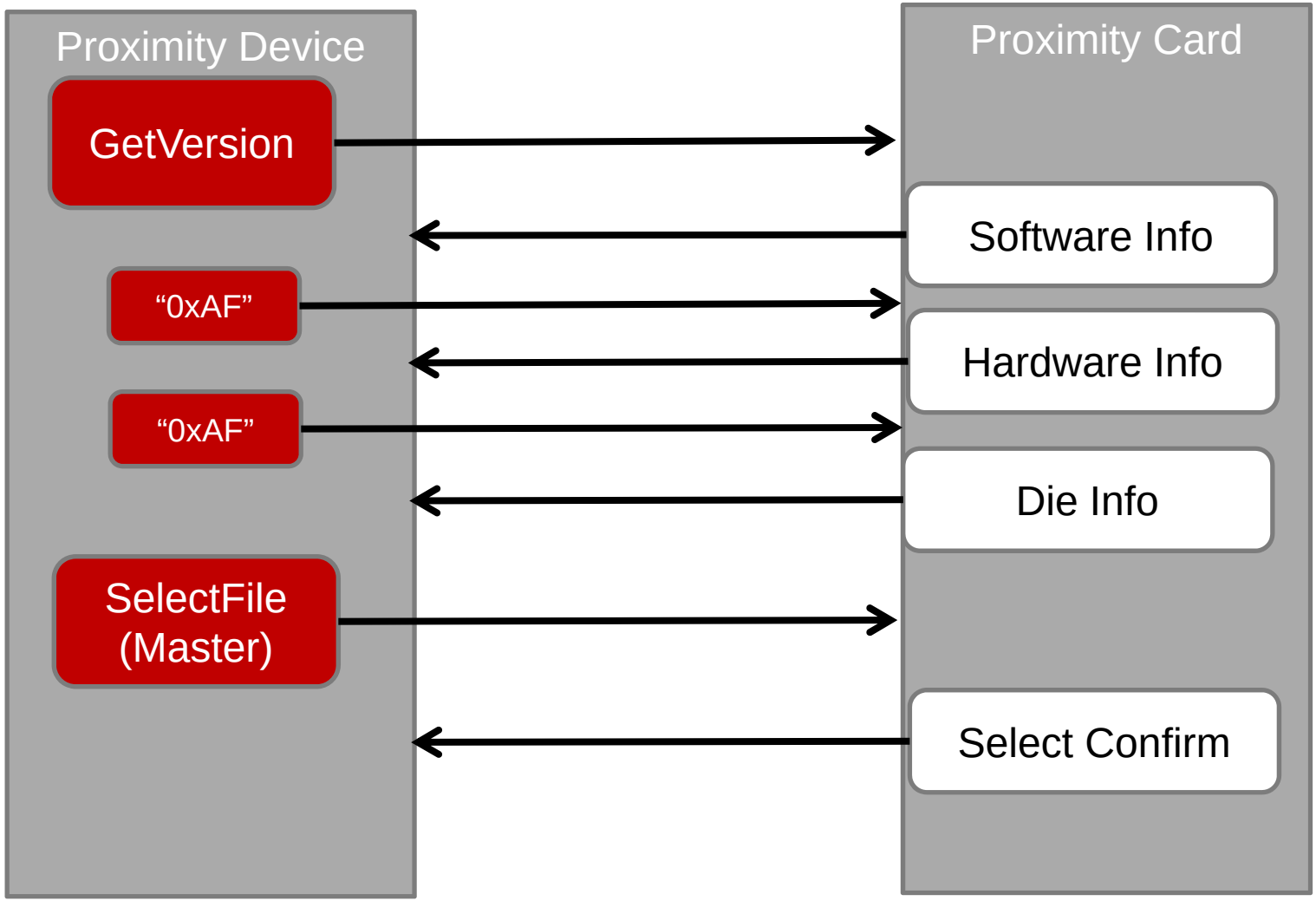
# Layer 4: Communication Setup

- Layer 4 is a separated function with own .c and .h files
- Current implementation is specific to MF DESFire EV1
  - Parameters easily changed with #defines



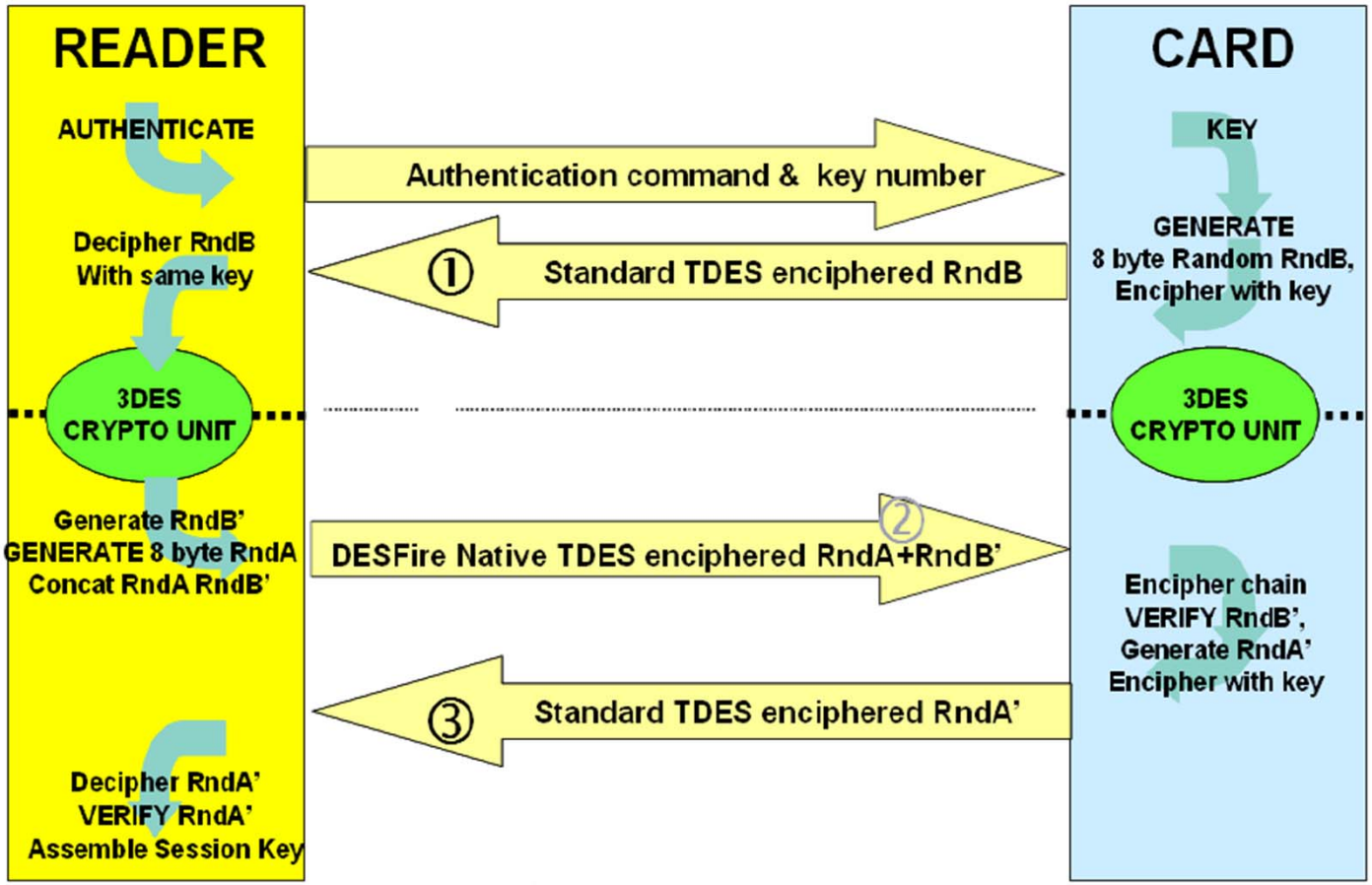
# DESFire EV1 Identification

- Separate .c and .h file; Performed by function call



# DESFire Authentication

- Separate .c and .h file; Performed by function call.




TDES Session Key: $RndA_{(byte\ 0\text{-}3)} + RndB_{(byte\ 0\text{-}3)} + RndA_{(byte\ 4\text{-}7)} + RndB_{(byte\ 4\text{-}7)}$

AES Session Key: $RndA_{(byte\ 0\text{-}3)} + RndB_{(byte\ 0\text{-}3)} + RndA_{(byte\ 12\text{-}15)} + RndB_{(byte\ 12\text{-}15)}$

# MFDESFIRE AES Authenticate

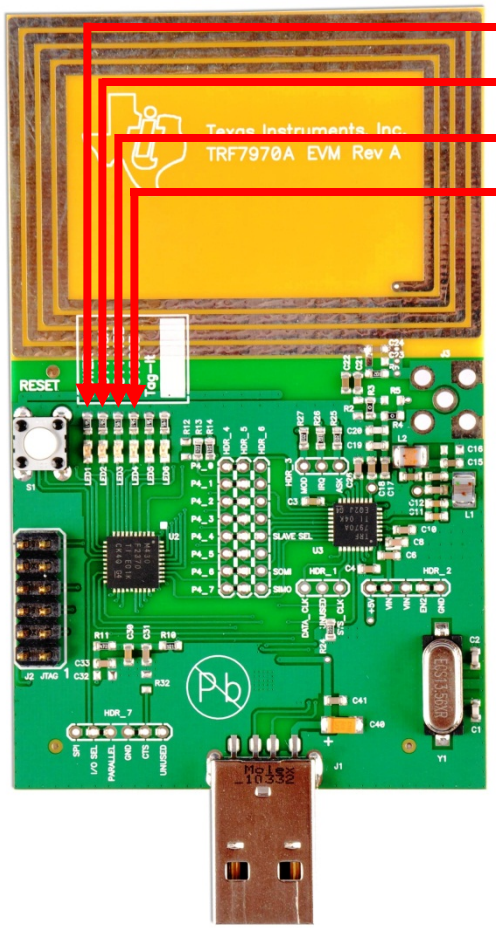
- AES_AUTHENTICATE( ) Parameters
  - u08_t *SessionKey
  - u08_t *RndA
  - u08_t *IV
  - u08_t *key

Extern pointers defined at main.c

# Issue: DES Code on LaunchPad

- DES Code too big for G2553 Launchpad.
- *Solution*:
  - Use TRF7970 EVM for development
  - Use AES encryption
    - Code and RAM size is smaller and offers better encryption
- *Process:*
  - Personalize card using DES to operate with AES
  - Develop CBC mode for DES code
  - Add CRC32 functionality

# AES Authentication Demo




- AES Demo on TRF7970AEVM
  - Green LED = Power
  - Red LED = Entering ISO14443a Anti-collision Loop
  - Yellow LED = Entering ISO14443a Layer 4
  - Orange LED = Successful AES Authentication

- Programmed EVM is powered from USB connection, runs in standalone, with card that has been configured for AES mode, using default keys

# Customer Actions for Use

- Random Number Generation
- CRC 32 Calculations
- Write Application

# Improvements

- “Wait for Response” Code needs to be more robust
  - Current Demo uses old code
  - Evaluation of response code too quick
  - MCU Delays inserted currently to get correct timing
- Offload Customer Actions to Native Support
  - Add Random Number Generation
  - Add CRC32 Functionality

# Wrapping Up

- We  have fully functional demo of AES Authentication with DESFIRE EV1.
- Demo can be configured for DES or TDES also.

# Back-Up Slides

# ISO14443 Layers 3 & 4

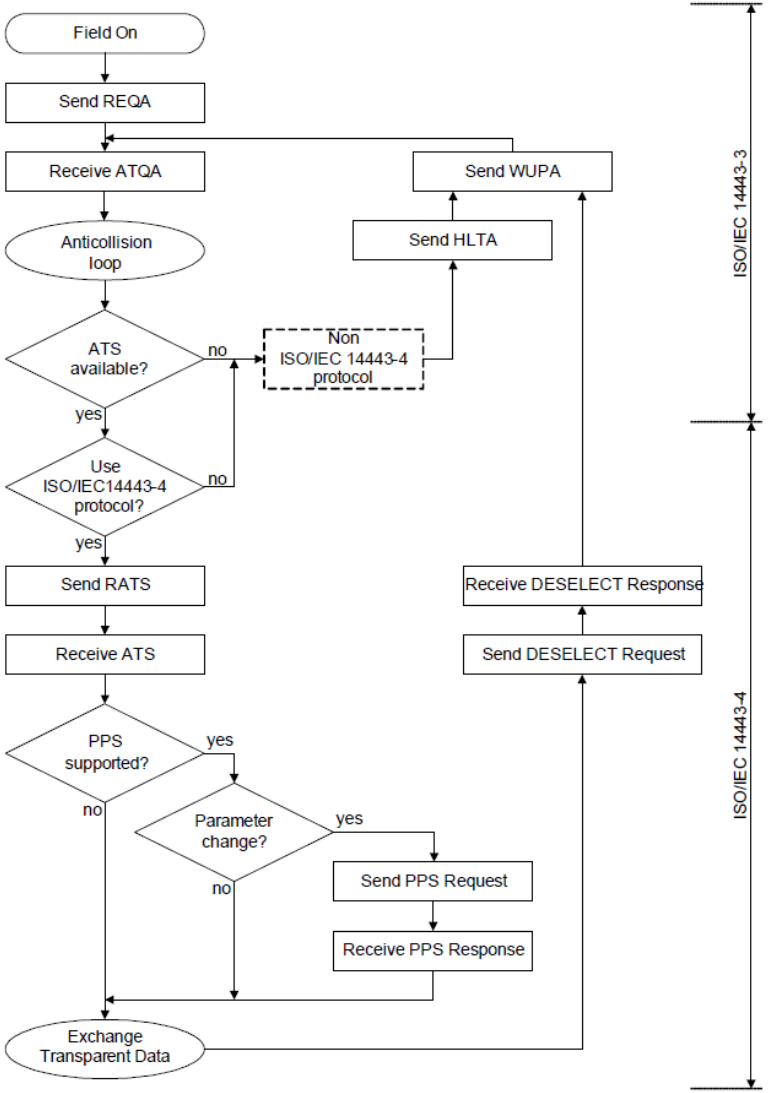
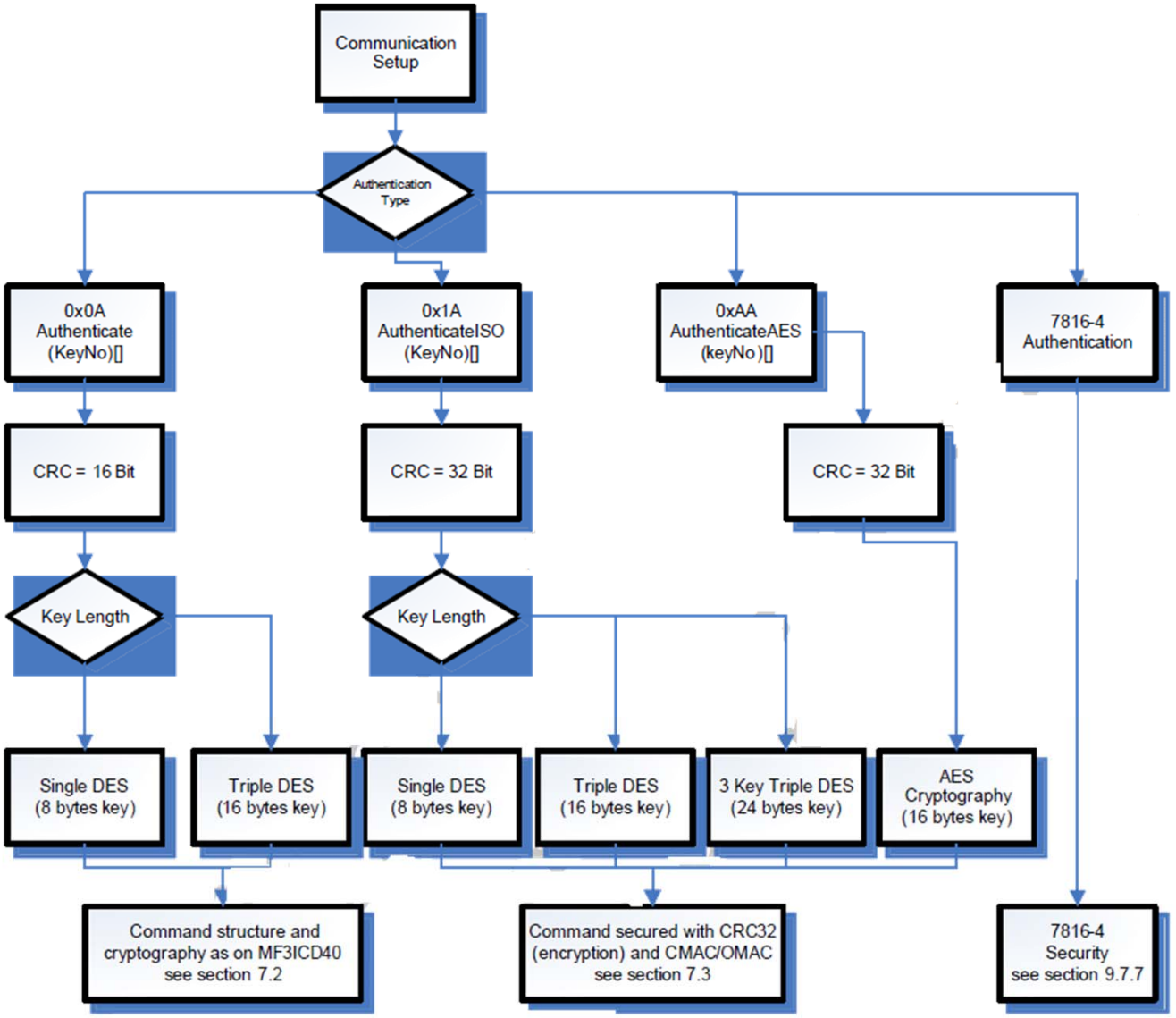



**Figure 1 — Activation of a PICC Type A by a PCD**

# MFDESFire EV1 Auth Identification



# Command Framing

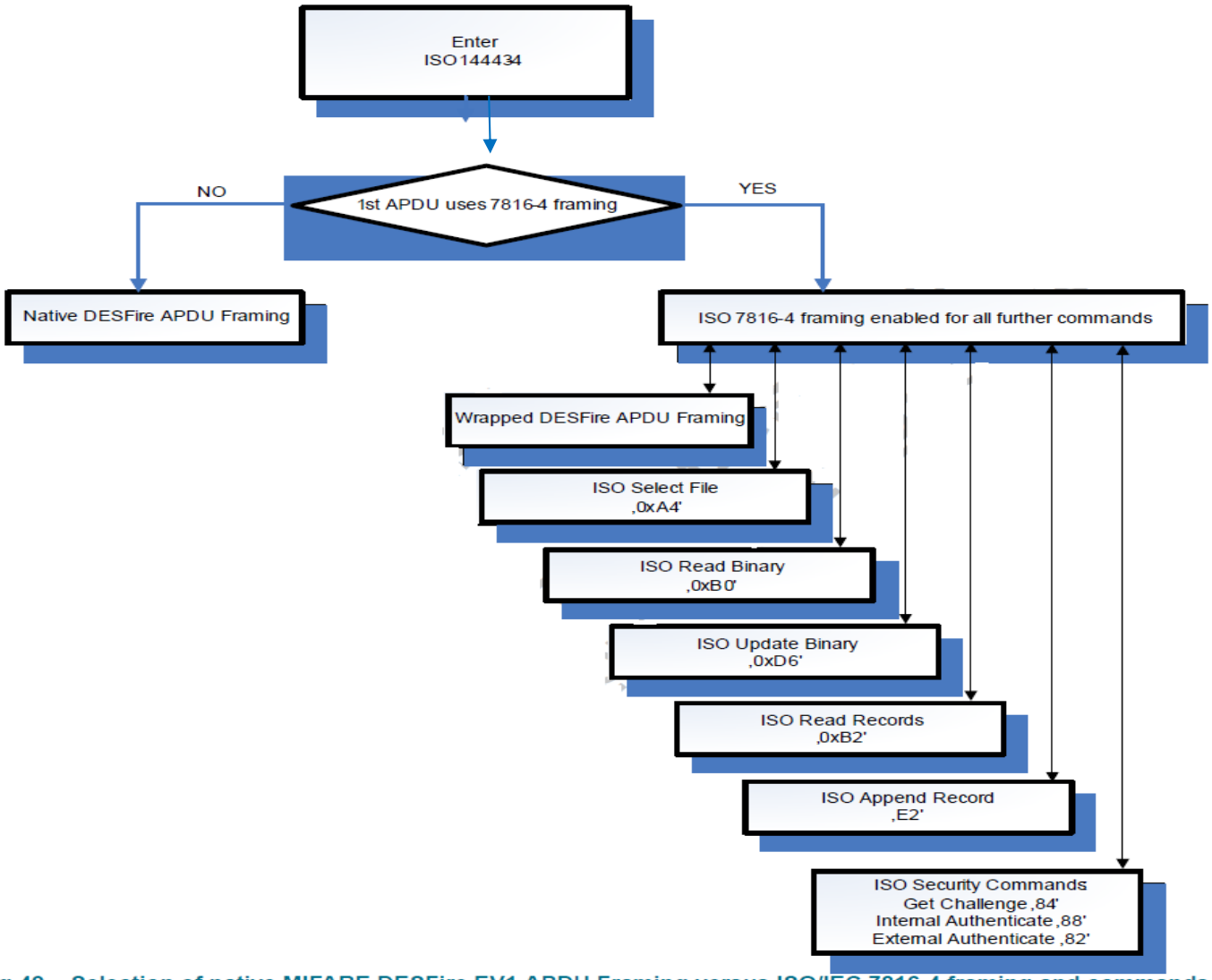



Fig 49. Selection of native MIFARE DESFire EV1 APDU Framing versus ISO/IEC 7816-4 framing and commands

# Crypto Code Benchmarks

| DES Code Size | Optimization | |
|---|---|---|
| | Speed | Size |
| Data (B) | 288 | 288 |
| Const (kB) | 2.3 | 2.3 |
| Code (kB) | 3.3 | 2.17 |

| DES Clock Cycle Count (kilo-cycles) | Optimization | |
|---|---|---|
| | Speed | Size |
| DES (FULL) (One Data Block) | 41 | 42.6 |
| 3DES (FULL) (One Data Block) | 135.6 | 143.1 |
| DES Key Schedular (EN0 or DE1 modes) | 34.7 | 36 |
| DES Key Schedular (ENDE mode) | 69 | 72 |
| DES Encode/Decode (One Data Block) | 2.7 | 3.8 |
| DES CBC Encode/Decode (2-block chain) | 5.5 | 7.7 |
| 3DES CBC Encode/Decode (2-block chain) | 139 | 149.7 |

| AES (ENC/DEC function) | | Optimization | |
|---|---|---|---|
| | | Speed | Size |
| Memory (kB) | Data (B) | 80 | 80 |
| | Const | 0.55 | 0.55 |
| | Code | 1 | 0.83 |
| Clock Cycles (kilo-cycles) | | 7.9 | 12.3 |

# RATS, ATS, PPSS Structure

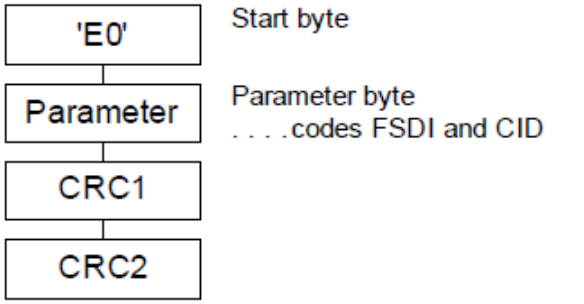



**Figure 2 — Request for answer to select**

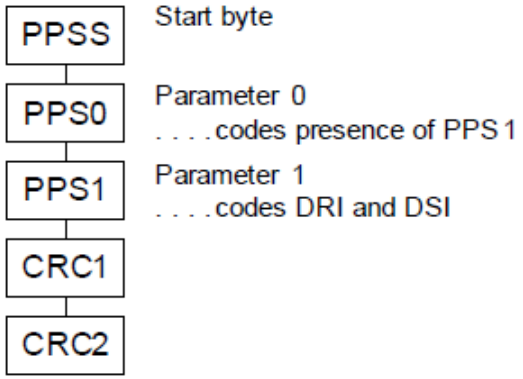



**Figure 9 — Protocol and parameter selection request**

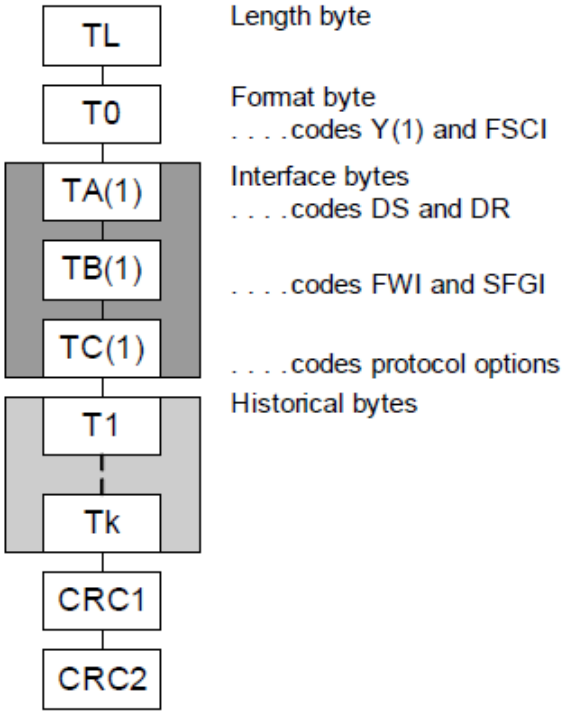



**Figure 4 — Structure of the ATS**

# RATS, ATS, PPSS Parameters

| | |
|---|---|
| FSDI | = Sets Frame Size (Device End) |
| CID | = Card Identification |
| FSCI | = Sets Frame Size (Card End) |
| TL | = Length Byte |
| Y(1) | = Presence of Interface Bytes |
| DS/DR | = Send/Receive Data rates |
| FWI | = Frame Waiting Interval |
| SFGI | = Safe Guard Interval |
| TC(1) | = Protocol Settings Supported |
| DSI/DRI | = Send/Receive Data rate settings |

# Anti-collision Loop

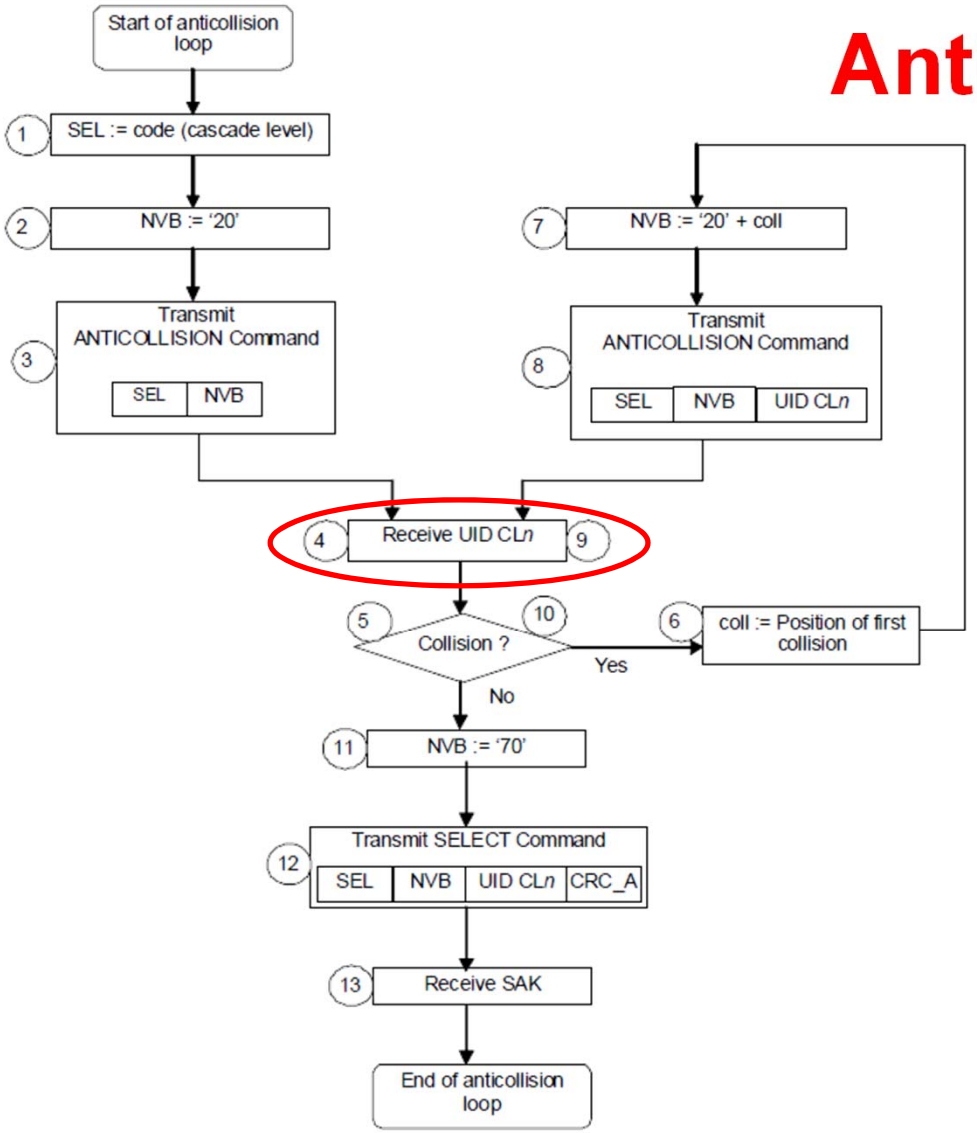



**Figure 10 — Anticollision loop, flowchart for PCD**

# Personalization

- After authentication, use ChangeKey command to switch to AES mode
  - Key = 16bytes of ‘0’, CRC32 needed over Command + Parameter + key
  - “0xC4800000000000000000000000000000008BE9EDB5000000”
  - Command+ Key#+ Key+ New Key Version + CRC32 +3 byte pad
  - Encrypt from Key to end of pad w/ SessionKey

“0xC4800000000000000000000000000000008BE9EDB5000000”