

A recent open source embedded implementation of the DESFire specification designed for on-the-fly logging with NFC based systems

Maxie Dion Schmidt

Georgia Institute of Technology
School of Mathematics
Atlanta, GA 30318, USA

mschmidt34@gatech.edu

<http://people.math.gatech.edu/~mschmidt34>

<https://github.com/maxieds>

Future Technologies Conference
October 2021

A recent open source embedded implementation of the
DESFire specification designed for on-the-fly logging
with NFC based systems

Maximilian Schmidt

Georgia Institute of Technology
School of Mathematics
Atlanta, GA 30332, USA
mschmidt@gatech.edu
<http://people.math.gatech.edu/~mschmidt14>
<https://github.com/mxscode>

Future Technologies Conference
October 2021

1. No notes for this page

High-level overview

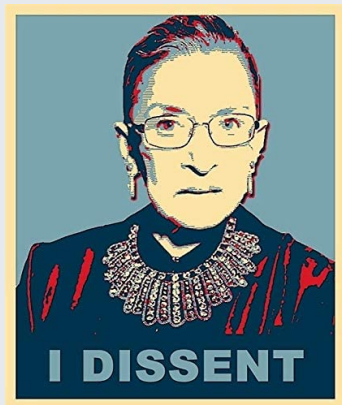
- ▶ Near Field Communication (NFC) protocol over short-distance RFID @ 13.56MHz
- ▶ Enables contactless data exchanges between passive tags (PICC) and active hosts (PCD)
- ▶ DESFire type cards provide modern cryptographic algorithms and more sophisticated feature set
- ▶ Chameleon Mini (RevG) devices used for pentesting and security applications as tag emulators and data loggers

- ▶ Near Field Communication (NFC) protocol over short-distance RFID @ 13.56MHz
- ▶ Enables contactless data exchanges between passive tags (PICC) and active hosts (PCD)
- ▶ DESFire type cards provide modern cryptographic algorithms and more sophisticated feature set
- ▶ Chameleon Mini (RevG) devices used for pentesting and security applications as tag emulators and data loggers

1. NFC used for wireless communication within a proximity of approximately 10 centimeters
2. Common in applications like physical authentication with door readers, university ID cards, to exchange credentials renting bikes or motorized scooters, and to charge limited credit transactions to vending machines and other virtual payment kiosks
3. Often encountered tag types include: MIFARE Classic, MIFARE Ultralight, NTAG and others over standardized ISO protocols and wrapped instruction sets
4. DESFire tags: DES/3DES/AES crypto + integrity checking, larger memory storage sizes (typically upto 8Kb), more complex filesystem organization and file types support, including secret key storage and variable access permissions
5. Chameleon Mini: A pentesting / development / security type device developed over the past six+ years or so that is designed to emulate common contactless tags, facilitate on-the-fly bitwise data exchanges, and log otherwise transparent low-level data exchanges over NFC

High-level overview (cont'd)

- ▶ DESFire emulation support for the Chameleon Mini has been a frequently requested, however complicated to deliver, feature for years
- ▶ How the first testing releases came together in the Fall of 2020
- ▶ <https://github.com/emsec/ChameleonMini/pull/287>



└ Introduction

└ High-level overview (cont'd)

High-level overview (cont'd)

- ▶ DESFire emulation support for the Chameleon Mini has been a frequently requested, however complicated to deliver, feature for years
- ▶ How the first testing releases came together in the Fall of 2020
- ▶ <https://github.com/ensec/ChameleonMini/pull/287>



1. The Chameleon RevG generation of devices includes hardware support on the integrated ATmega chip for AES and DES cryptographic primitives.
2. Even still only partial support for DESFire tags was added as a fork of the main Chameleon firmware by @devzzo in 2017
3. I decided to take on the task publicly ironically drinking beer after reading another issue requesting this support on the main firmware sources page on *GitHub* in the Spring of 2020
4. I am proud to have been able to do what I characterize as having “hacked for freedom with free and open source software” to bring awareness to certain societal issues and systematic injustice during the first wave of the COVID-19 pandemic in 2020. In fact, the first public testing release of the DESFire enabled firmware binaries was made after the low-level source code came together with me staying up the entire weekend with vim on my Mac terminal after late US Supreme Court Justice, RBG, sadly passed from pancreatic cancer on Rosh Hashanah last year. The release notes included an iconic image of her reading “*I dissent*”.

High-level overview (cont'd)

- ▶ Significance: First of its kind functional embedded proof-of-concept DESFire stack that is freely available as OSS to researchers, security experts and end users alike
- ▶ Limitations: Small R&D budget for testing and lack of standardized default data transfer modes to ensure interoperability amongst door readers in applications

- ▶ Significance: First of its kind functional embedded proof-of-concept DESFire stack that is freely available as OSS to researchers, security experts and end users alike
- ▶ Limitations: Small R&D budget for testing and lack of standardized default data transfer modes to ensure interoperability amongst door readers in applications

1. No notes for this page

Outline of topics

2021-08-27

FTC 2021 — Embedded DESFire

└ Introduction

[Outline of topics](#)

1. No notes: Title slide only

Presentation outline

- ▶ The Chameleon Mini device hardware profile and embedded software features
- ▶ Overview of key features of the proprietary DESFire command set
- ▶ Key features and challenges in writing the embedded DESFire implementation (with examples)

2021-08-27

FTC 2021 — Embedded DESFire

└ Introduction

└ Presentation outline

Presentation outline

- ▶ The Chameleon Mini device hardware profile and embedded software features
- ▶ Overview of key features of the proprietary DESFire command set
- ▶ Key features and challenges in writing the embedded DESFire implementation (with examples)

1. No notes for this page

Chameleon Mini Hardware

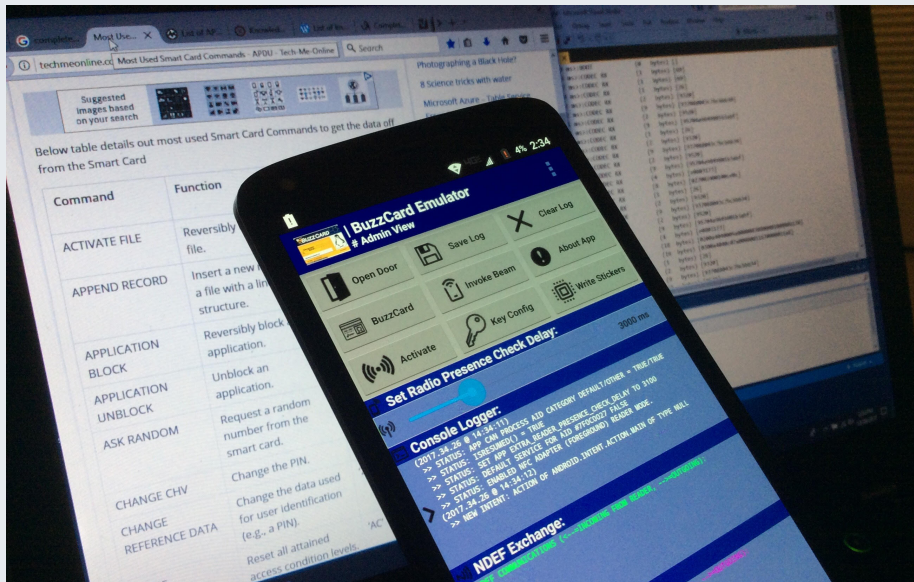
2021-08-27

FTC 2021 — Embedded DESFire
└ Chameleon Mini Hardware

Chameleon Mini Hardware

1. No notes: Title slide only

Origins of the project I



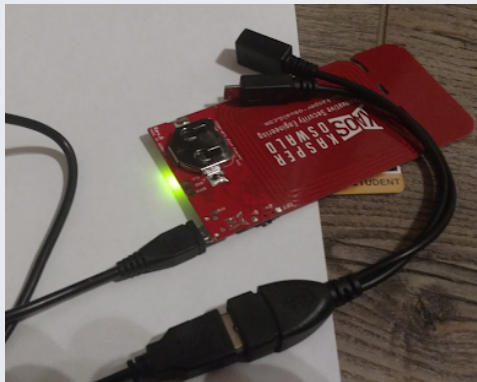
FTC 2021 — Embedded DESFire

- Chameleon Mini Hardware
 - Motivation
 - Origins of the project I



1. Moved to GA Tech in 2017 as a Ph.D. student in the School of Math
2. Shortly after arriving on campus I was issued a student ID with an integrated DESFire EV1 tag
3. This was also around the time I had purchased my first developer grade Android phone
4. I decided that I wanted the physical authentication to doors on campus to work not only with the standard issue university ID but also with my phone
5. Exploration with Android OS application development and limitations of low-level NFC data exchange transparency on the stock MotoDroid led me to seek external hardware to help reverse engineer the bytes I would need to exchange from phone to door reader, and vice versa

Origins of the project II



Chameleon Mini Live Debugger
Portable logging interface v0.1.1

MF_CLASSIC_4K MEM-4K/LOG-2K/DIP#1
AB:CD:EF:01 RW/FLD-0/NO-CHRG
THRS-400 mv/TMF-5000 ms

Log Tools Menu Log Tools Export

000006 LOG_CODE_DNE 4B +0ms
00 00 00 00
... ISO_STD_APOU

000007 LOG_CODE_DNE 4B +0ms
00 00 00 00
... ISO_STD_APOU

000008 LOG_CODE_DNE 4B +0ms
00 00 00 00
... ISO_STD_APOU

000009 CONFIG_SET 24B ~5656ms
49 53 4f 31 34 34 34 33 41 5f 52 45 41 44 45 52 31 30 30 3a 4f 4b 0d 0a
ISO14443A_READER100-OK...

000010 CONFIG_SET 21B +24946ms
4d 46 5f 43 4c 41 53 53 49 43 5f 34 4b 31 30 30 3a 4f 4b 0d 0a
MF_CLASSIC_4K100-OK
GENERATE_PUBLIC_KEY_PAIR (ISO 7816-8)

INFO: SHELL COMMAND OF UID=ABCDEF01
RETURNED STATUS **K** 100:OK --
2018-01-11-20:30:14

2021-08-27

FTC 2021 — Embedded DESFire

- Chameleon Mini Hardware
 - Motivation
 - Origins of the project II

Origins of the project II



1. Enter the KAOS Chameleon Mini RevG devices from the EU
2. One of the key new features of the RevG generation of Chameleon devices is a LIVE logging feature by which the device can sniff bidirectional NFC data packets and print them in realtime over the integrated serial USB interface
3. The RHS image shows an early prototype of my CMLD application for Android designed to display this LIVE logging data in human readable format portably and on-the-fly

The Chameleon Mini device profile (hardware)

- ▶ On-board integration of a modern AVR chip (ATxmega128A4U)
- ▶ Memory: 128Kb of FLASH, 8Kb of SRAM, and 2Kb of EEPROM spaces and support for faster FRAM-based memory access
- ▶ Accelerated hardware support for AES and DES cryptographic engines
- ▶ Embedded firmware and flashable bootloader support to memory map the integrated RF hardware on the PCB
- ▶ Serial data transfer over wired micro-USB

- ▶ On-board integration of a modern AVR chip (ATxmega128A4U)
- ▶ Memory: 128Kb of FLASH, 8Kb of SRAM, and 2Kb of EEPROM spaces and support for faster FRAM-based memory access
- ▶ Accelerated hardware support for AES and DES cryptographic engines
- ▶ Embedded firmware and flashable bootloader support to memory map the integrated RF hardware on the PCB
- ▶ Serial data transfer over wired micro-USB

1. No notes for this page

The Chameleon Mini device profile (software)

- ▶ Embedded OSS firmware and bootloader sources in C and ASM compiled with `avr-gcc` that are flashed to the device over USB
- ▶ Convenient serial terminal that has a human-readable command set for easy on-the-fly configuration of emulated tags
- ▶ Ability to act as a PICC, PCD or bidirectional NFC packet sniffer depending on the active configuration set in one of the eight 8Kb sized partitions of the onboard memory
- ▶ Logging of time-stamped communication details and status events to internal FRAM memory or LIVE mode printed to the serial USB

- ▶ Embedded OSS firmware and bootloader sources in C and ASM compiled with `avr-gcc` that are flashed to the device over USB
- ▶ Convenient serial terminal that has a human-readable command set for easy on-the-fly configuration of emulated tags
- ▶ Ability to act as a PICC, PCD or bidirectional NFC packet sniffer depending on the active configuration set in one of the eight 8Kb sized partitions of the onboard memory
- ▶ Logging of time-stamped communication details and status events to internal FRAM memory or LIVE mode printed to the serial USB

1. No notes for this page

DESFire tags

2021-08-27

FTC 2021 — Embedded DESFire

└─ DESFire NFC Tags

DESFire tags

1. No notes: Title slide only

Key Features

- ▶ Multiple nested and semi-interoperable generations of DESFire tags: Legacy Mifare DESFire, EV1, EV2, EV3 and Light variants
- ▶ Larger scale integrated memory storage sizes than most contactless NFC tags (usually 2Kb, 4Kb or 8Kb)
- ▶ Standard use of modern cryptographic algorithms for secure data exchange (legacy DES/3DES/AES-128/AES-256)
- ▶ Data messages optionally padded with cryptographically hashed bytes to ensure data integrity over the physical interface using 2-byte CRC checksums or 4-byte MAC trailers
- ▶ Implementations are complicated by proprietary handling of most DESFire tag specs by the manufacturers

- ▶ Multiple nested and semi-interoperable generations of DESFire tags: Legacy Milan DESFire, EV1, EV2, EV3 and Light variants
- ▶ Larger scale integrated memory storage sizes than most contactless NFC tags (usually 2Kb, 4Kb or 8Kb)
- ▶ Standard use of modern cryptographic algorithms for secure data exchange (Legacy DES/3DES/AES-128/AES-256)
- ▶ Data messages optionally padded with cryptographically hashed bytes to ensure data integrity over the physical interface using 2-byte CRC checksums or 4-byte MAC trailers
- ▶ Implementations are complicated by proprietary handling of most DESFire tag specs by the manufacturers

1. No notes for this page

Filesystem: Organization and internal storage types

- ▶ Files grouped by allocations of the physical IC memory into top-level subdirectories called applications indexed by unique application identifier (AID)
- ▶ Native file types: Standard data files (type 0), backup data files (type 1), value files (type 2), linear record files (type 3), and cyclic record files (type 4)
- ▶ Each file has 2-bytes of associated access rights to indicate one of read/write/read and write/change.
- ▶ Access permissions on the files provide more secure protections for storage of secret binary key data

- ▶ Files grouped by allocations of the physical IC memory into top-level subdirectories called applications indexed by unique application identifier (AID)
- ▶ Native file types: Standard data files (type 0), backup data files (type 1), value files (type 2), linear record files (type 3), and cyclic record files (type 4)
- ▶ Each file has 2-bytes of associated access rights to indicate one of read/write/read and write/change.
- ▶ Access permissions on the files provide more secure protections for storage of secret binary key data

1. There is a default master (PICC) application with associated master keys for authentication that is the default selected AID upon initial handshaking from PICC to PCD and vice versa
2. The actively selected AID can be changed via another subsequent structured command call initiated from PICC to PCD
3. Within each application space, the file entries are partitioned into data files or records that can store variable length hexadecimal-formatted binary data or signed integer values that can be debited and credited by invoking native instructions
4. Access to sensitive files secured by the cryptographic mechanisms supported by these tags requires both a base round of initial handshaking (PICC-to-PCD) that generates a session key, which is then followed by a cryptographic checksum verified exchange of the authentication process using a secret DES/3DES/AES key

Commands and native instruction support

- ▶ Formats to communicate instructions is performed by sending unpadded native commands or by communicating ISO standardized wrapped APDU messages

PICC-to-PCD wrapped APDU data exchange format:

CLA	INS	P ₁	P ₂	L _c	Data Bytes	L _e
0x90	command code	0x00	0x00	variable length of data	command data	0x00

PCD-to-PICC format:

Data Bytes	SW1	SW2 (Status)
DESFire command response data	0x91	0xYY

► Formats to communicate instructions is performed by sending unpadded native commands or by communicating ISO standardized wrapped APDU messages

PICC-to-PCD wrapped APDU data exchange format:

CLA	INS	P1	P2	Lc	Data bytes	Lr
0x00	command code	0x00	0x00	variable length of data	command data	0x00

PCD-to-PICC format:

SW1	SW2 (Status)
0x00	0x00
DESFire command response data	0x01 0x01

1. Wrapped APDU format for native DESFire commands in the PICC-to-PCD direction (ISO-7816-5 message structure).
2. Format of the response message for native DESFire commands in the PCD-to-PICC direction. The SW2 status code byte returned by the PCD (denoted by $0xYY$ above) is set to either $0x00$ to indicate no error in processing the command or is encoded as a reserved byte code to provide an explanation of an error that occurred on the PCD side. The returned error codes are used to indicate problems ranging from hardware errors, to authentication and access permissions errors, to AID and file not found warnings, or to communicate that invalid parameters were passed in the issuing command call.

Supported command codes

Command Long Name	INS	Description
AUTHENTICATE	0x0A	Legacy mode authentication
AUTHENTICATE_ISO	0x1A	ISO authentication with 3DES
AUTHENTICATE_AES	0xAA	Standard AES authentication
AUTHENTICATE_EV2_FIRST	0x71	More recent EV2 authentication mode
AUTHENTICATE_EV2_NONFIRST	0x77	More recent EV2 authentication mode
CHANGE_KEY_SETTINGS	0x54	Modify PICC master key properties
SET_CONFIGURATION	0x5C	Used to configure DESFire card or application specific attributes
CHANGE_KEY	0xC4	Changes the key data stored on the PICC
GET_KEY_VERSION	0x64	Returns the active key version stored on the PICC
CREATE_APPLICATION	0xCA	Creates new applications by unique AID
DELETE_APPLICATION	0xDA	Non-restorable deletion operation
GET_APPLICATION_IDS	0x6A	Returns a list of all AID codes stored on the PICC
FREE_MEMORY	0x6E	Returns the total free memory on the tag in bytes
GET_DF_NAMES	0x6D	Obtain the ISO7816-4 DF names associated with the tag
GET_KEY_SETTINGS	0x45	Get permissions data and format for PICC and application master keys
SELECT_APPLICATION	0x5A	Select a specific application by AID for further access

DESFire NFC Tags

Supported command codes

Supported command codes

Command Long Name	ID#	Description
AUTHENTICATE	0x00	Legacy mode authentication
AUTHENTICATE_100	0x10	ISO authentication with 3DES
AUTHENTICATE_400	0x40	Standard AES authentication
AUTHENTICATE_401/FIRST	0x71	More secure EV2 authentication mode
AUTHENTICATE_402/CONFIRST	0x77	More secure EV2 authentication mode
CHANGE_KEY_SETTINGS	0x04	Modify PCCC master key properties
SET_CONFIGURATION	0x0C	Used to configure DESFire card or application specific attributes
CHANGE_KEY	0x04	Changes the key data stored on the PCCC
SET_KEY_VERSION	0x04	Returns the active key version stored on the PCCC
CREATE_APPLICATION	0x04	Creates new applications by unique AID
DELETE_APPLICATION	0x00	Non-reversible deletion operation
GET_APPLICATIONS_LIST	0x04	Returns a list of all AID codes stored on the PCCC
FREE_MEMORY	0x00	Returns the total free memory on the tag in bytes
GET_P_SETTINGS	0x00	Obtain the ICCP/SAFE DF names associated with the TAG
SET_KEY_SETTINGS	0x04	Get permissions data and format for PCCC and application master keys
SELECT_APPLICATION	0x04	Select a specific application by AID for further access

1. Commands to authenticate with a few cryptographic protocols (e.g., legacy mode, 3DES, AES), modify and create keys, and create container applications (much like directories)

Supported command codes

Command Long Name	INS	Description
FORMAT_PICC	0xFC	Releases the previously stored user memory (not reversible)
GET_VERSION	0x60	Returns manufacturing header data stored in the PICC
GET_CARD_UID	0x51	Returns the 7-byte card UID assigned by the manufacturer
GET_FILE_IDS	0x6F	Get a list of the file identifiers (by index) within the selected AID
GET_FILE_SETTINGS	0xF5	Obtain properties and permissions about a file
CHANGE_FILE_SETTINGS	0x5F	Modify access permissions of an existing file
CREATE_STDDATA_FILE	0xCD	Add new unformatted binary data storage file type
CREATE_BACKUPDATA_FILE	0xCB	Create unformatted binary file with a shadow backup mechanism
CREATE_VALUE_FILE	0xCC	Create new 32-bit integer storage file
CREATE_LINEAR_RECORD_FILE	0xC1	Create new fixed size file for sequential storage of structurally similar record data structures
CREATE_CYCLIC_RECORD_FILE	0xC0	Similar to the linear record case except that there is a wrap-around storage functionality when the file size limit is exceeded

DESFire NFC Tags

Supported command codes

Supported command codes

Command Long Name	ID	Description
FORMAT_FILE	0x01	Initialize the previously stored user memory (not re-writable)
GET_MFGID	0x48	Returns manufacturing header data stored in the PUID
GET_UID	0x49	Returns the 7-byte card UID assigned by the manufacturer
GET_FILE_IDN	0x4F	Get a list of the file identifiers (by index) within the selected AID
GET_FILE_SETTINGS	0x51	Obtain properties and permissions about a file
MODIFY_FILE_SETTINGS	0x5F	Modify access permissions of an existing file
CREATE_UNFORMATTED_FILE	0x2D	Add new unformatted binary data storage file type
CREATE_BACKUP_FILE	0x2B	Create unformatted binary file with a shadow backup mechanism
CREATE_IMAGE_FILE	0x2C	Create new 3D-bit image storage file
CREATE_LINEAR_RECORD_FILE	0x2E	Create new fixed size file for sequential storage of records
CREATE_WRAP_AROUND_FILE	0x2F	Linearly similar record data structure Similar to the linear record case except that there is a wrap-around storage functionality when the file size limit is exceeded

1. Commands to reset the tag to a default blank contents mode, obtain the manufacturer bytes, and create files of various types

Supported command codes

Command Long Name	INS	Description
DELETE_FILE	0xDF	Non-restorable deactivation of a file within the active AID
GET_ISO_FILE_IDS	0x61	Returns a list of the 2-byte file identifiers of all files within the active AID
READ_DATA	0xBD	Read byte-wise contents of standard or backup file types
WRITE_DATA	0x3D	Write data at an offset to standard or backup file types
GET_VALUE	0x6C	Reads the last permanently stored integer from value records
CREDIT	0x0C	Increase the integer value type in the value type
DEBIT	0xDC	Decrease the integer value type in the value type
LIMITED_CREDIT	0x1C	Increase by a preset limited amount the integer in a value record (must commit the transaction changes at a later time)
WRITE_RECORD	0x3B	Write data to a linear or cyclic record file type
READ_RECORDS	0xBB	List the set of complete records in the associated file type
CLEAR_RECORD_FILE	0xEB	Reset a linear or cyclic record type to an empty state
COMMIT_TRANSACTION	0xC7	Validates the previous write access permissions and credit permissions of all files within the selected AID

DESFire NFC Tags

Supported command codes

Supported command codes

Command	Long Name	RES	Description
0x11	GET_AID	0x00	Non-recursion destruction of a file within the active AID
0x12	GET_AIDS	0x01	Returns a list of the 3-byte file identifiers of all files within the active AID
0x13	READ_DATA	0x00	Read byte-wise contents of standard or backup file types
0x14	WRITE_DATA	0x00	Write data at an offset to standard or backup file types
0x15	GET_VALUE	0x00	Reads the last permanently stored integer from value records
0x16	INCREMENT	0x00	Increase the integer value type in the value type
0x17	DECREMENT	0x00	Decrease the integer value type in the value type
0x18	LIMITED_INCREMENT	0x00	Increase by a preset limited amount the integer in a value record (must cannot the transaction changes at a later time)
0x19	WRITE_RECORD	0x00	Write data to a linear or cyclic record file type
0x1A	READ_RECORDS	0x00	List the set of complete records in the associated file type
0x1B	CREATE_RECORDFILE	0x00	Reset a linear or cyclic record type to an empty state
0x1C	COMMIT_TRANSACTION	0x01	Validates the previous write access permissions, and credit permissions of all files within the selected AID

1. Commands to delete files and read/write/modify their respective contents

Supported command codes

Command Long Name	INS	Description
ABORT_TRANSACTION	0xA7	Invalidate the previous changes to the files within the selected AID
SELECT	0xA4	ISO7816-4 standard command support
GET_CHALLENGE	0x84	ISO7816-4 standard command support
EXTERNAL_AUTHENTICATE	0x82	ISO7816-4 standard command support
INTERNAL_AUTHENTICATE	0x88	ISO7816-4 standard command support
READ_BINARY	0xB0	ISO7816-4 standard command support
UPDATE_BINARY	0xD6	ISO7816-4 standard command support
READ_RECORDS	0xB2	ISO7816-4 standard command support
APPEND_RECORD	0xE2	ISO7816-4 standard command support

└─ DESFire NFC Tags

└─ Supported command codes

Command Long Name	ISO	Description
SELECT_NOTIFICATION	ISO4	incorporates the previous change to the file within the selected AID
SELECT	ISO4	ISO7816-4 standard command support
SET_CHALLENGE	ISO4	ISO7816-4 standard command support
EXTENDED_AUTHENTICATE	ISO4	ISO7816-4 standard command support
EXTENDED_AUTHENTICATE	ISO4	ISO7816-4 standard command support
READ_BINARY	ISO4	ISO7816-4 standard command support
WRITE_BINARY	ISO4	ISO7816-4 standard command support
READ_RECORDS	ISO4	ISO7816-4 standard command support
WRITE_RECORDS	ISO4	ISO7816-4 standard command support

1. A subset of the ISO7816-4 standard commands

Data exchanges with the Chameleon DESFire configuration

```
>>> Select Application By AID:
-> 90 5a 00 00 03 00 00 00 | 00
<- 91 00

>>> Start AES Authenticate:
-> 90 aa 00 00 01 00 00
<- 54 b8 9e fe 19 9b c6 a5 | fd 8f 00 be c1 23 99 c0 | 91 af
-> 90 af 00 00 10 df a0 79 | 13 59 ac 4c 75 5f 81 69 |
   bc 9c 3e c6 7e 00
<- a9 e2 79 42 11 63 9c 14 | 07 b3 02 2f 2e 4b 2e c5 | 91 00

>>> Get AID List From Device:
-> 90 6a 00 00 00 00
<- 77 88 99 01 00 34 91 00

>>> CreateApplication command:
-> 90 ca 00 00 05 77 88 99 | 0f 03 00
<- 91 de

>>> Get AID List From Device:
-> 90 6a 00 00 00 00
<- 77 88 99 01 00 34 91 00
```

```

1 >>> select application by AID:
2   -> 00 00 00 00 00 00 00 00 | 00
3   <- 00 00
4
5 >>> Start AID's authentication:
6   -> 00 00 00 00 00 00 00 00
7   -> 00 00 00 00 00 00 00 00 | 00 00 00 00 c1 23 00 00 | 01 0f
8   -> 00 00 00 00 00 00 00 00 | 13 00 00 00 00 00 00 00 |
9     00 00 00 00 00 00
10  -> 00 00 78 43 11 03 0c 14 | 03 02 02 2f 2a 4b 2a c5 | 01 00
11
12 >>> Get AID list from device:
13   -> 00 00 00 00 00 00
14   <- 77 00 00 01 00 24 01 00
15
16 >>> Create application command:
17   -> 00 00 00 00 00 77 00 00 | 0f 03 00
18   <- 00 00
19
20 >>> Get AID list from device:
21   -> 00 00 00 00 00 00
22   <- 77 00 00 01 00 24 01 00

```

1. More complete examples of data exchanges using these commands are found in the conference proceedings article and in the LibNFC testing code within the Chameleon mini main firmware repository on *GitHub*

An Embedded Open Source DESFire Stack for the Chameleon Mini

2021-08-27

FTC 2021 — Embedded DESFire

└─ OSS Embedded DESFire

An Embedded Open Source DESFire
Stack for the Chameleon Mini

1. No notes: Title slide only

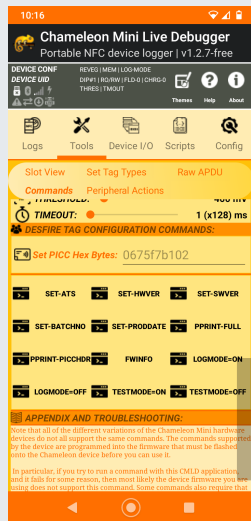
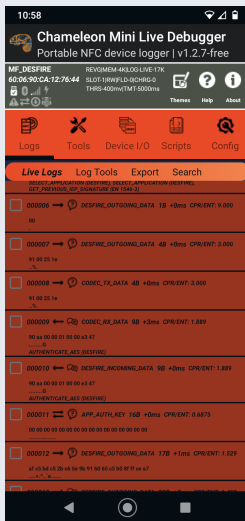
Extensions of the firmware sources to support DESFire tags

- ▶ New native AES support using hardware acceleration support
- ▶ Extensions of prior work to add hardware based DES and 3DES support to the firmware
- ▶ Changes to the codec layer of the firmware to support DESFire tags
- ▶ Enhancements and bug fixes to the LIVE logging functionality of the Chameleon RevG devices
- ▶ New default customized extension of the Chameleon terminal commands to enhance DESFire configuration support for users (see next slide)

Configuration of DESFire emulation support

```
> CONFIG=MF_DESFIRE
> DF_SETHDR=ATS 0675F7B102
> UID=2377000B99BF98
```

```
DF_SETHDR=ATS xxxxxxxxxxxx
DF_SETHDR=HardwareVersion xxxx
DF_SETHDR=SoftwareVersion xxxx
DF_SETHDR=BatchNumber xxxxxxxxxxxx
DF_SETHDR=ProductionDate xxxx
```



APPENDIX AND TROUBLESHOOTING:
 Note that all of the different variations of the Chameleon Mini hardware devices do not all support the same commands. The commands supported by the device are programmed into the firmware that will be flashed onto the Chameleon device before you can use it.

In particular, if you try to run a command with this CMDL application and it fails for some reason, then most likely the device firmware you are using does not support this command. Some commands also require that

```
> CONF1=HF_SICF3BC  
> DF_SETHDR=415 847578182  
> UID=127798899616
```

```
-----  
DF_SETHDR=415 xxxxxxxx  
DF_SETHDR=Manufacturer xxxx  
DF_SETHDR=GetDate=xxxx xxxx  
DF_SETHDR=ProductDate xxxxxxxx  
DF_SETHDR=ProductDate xxxx
```



1. Chameleon Mini terminal input for firmware compiled with DESFire emulation support to configure an IBM-JCOP branded NFC tag.
2. The extended Chameleon terminal for DESFire emulation can be used to clone the header data on the tag that is assumed unique by the manufacturer. This includes ATS bytes, HW/SW versions, batch numbers, the production date, and of course the UID that can be reset by all Chameleon tag configurations

DESFire emulation support (anti-collision loop)

NFC reader: SCM Micro / SCL3711-NFC&RW opened

```

Sent bits:      26 (7 bits)
Received bits: 03 44
Sent bits:      93 20
Received bits: 88 23 77 00 dc
Sent bits:      93 70 88 23 77 00 dc 4b b3
Received bits: 04
Sent bits:      95 20
Received bits: 0b 99 bf 98 b5
Sent bits:      95 70 0b 99 bf 98 b5 2f 24
Received bits: 20
Sent bits:      e0 50 bc a5
Received bits: 75 77 81 02 80
Sent bits:      50 00 57 cd
  
```

Found tag with

UID: 2377000b99bf98

ATQA: 4403

SAK: 20

ATS: 75 77 81 02 80

```
NFC reader: SCM Micro / SCL1711-NFCRW opened
Sent Bits: 26 (7 bits)
Received Bits: 63 69
Sent Bits: 63 28
Received Bits: 68 23 77 88 0c 4b 5a
Sent Bits: 63 78 88 23 77 88 0c 4b 5a
Received Bits: 64
Sent Bits: 6c 28
Received Bits: 6c 9f 98 85
Sent Bits: 6c 78 88 23 77 88 0c 4b 5a 2f 24
Received Bits: 28
Sent Bits: 68 58 6c 6c
Received Bits: 75 77 81 83 88
Sent Bits: 68 88 57 02

Found tag with
UID: 2270800000000000
ATQA: 4448
SAC: 28
ATA: 75 77 81 83 88
```

1. Output of reading the resulting DESFire tag emulated by the Chameleon Mini device using an externally connected USB NFC reader with the LibNFC nfc-anticol utility

Challenges with the implementation during development

- ▶ Approximately six to eight months of active development were required to complete the project
- ▶ Forced by local embedded system constraints to carefully optimize and organize our use of the embedded AVR memory to resolve insufficient memory type exceptions
- ▶ The speedup in computations for AES and 3DES operations provides an order of magnitude improvement compared to existing OSS libraries for AVR chips
- ▶ A complicated nested, quasi-linked pointer based structure was required to efficiently store the filesystem entries and tag accounting metadata

- ▶ Approximately six to eight months of active development were required to complete the project
- ▶ Forced by local embedded system constraints to carefully optimize and organize our use of the embedded AVR memory to resolve insufficient memory type exceptions
- ▶ The speedup in computations for AES and 3DES operations provides an order of magnitude improvement compared to existing OSS libraries for AVR chips
- ▶ A complicated nested, quasi-linked pointer based structure was required to efficiently store the filesystem entries and tag accounting metadata

1. Most notably, the structures and buffer space needed to store cryptographic structures for use with AES and 3DES were carefully leveraged on the stack to avoid unrecoverable overflow errors and race conditions
2. Primarily used LibNFC on MacOS and Linux to test the implementation with an external USB NFC tag reader/writer
3. The testing code in C was contributed to the firmware sources with the main PR to add the DESFire support
4. Several sample dumps of the working implementation are also bundled with the firmware repo to verify compatibility and functionality whenever new PRs are incorporated to extend the current working implementation. Some experimental non-default features are tested by editing the Makefile

Concluding Remarks

2021-08-27

FTC 2021 — Embedded DESFire

- └ Credits and Concluding Discussion

Concluding Remarks

1. No notes: Title slide only

Funding sources and support for the project

- ▶ Initial sources for the DESFire Chameleon firmware are due to Dmitry Janushkevich (**@devzzo**) (2017)
- ▶ Professor Josephine Yu in the School of Math at GA Tech in the US
- ▶ The original Kasper and Oswald (KAOS) developers of the Chameleon Mini hardware and software
- ▶ David Oswald from the University of Birmingham in the UK

Wrapping up: Selected quotes on open source software

That brings me to the most important piece of advice that I can give to all of you: if you've got a good idea, and it's a contribution, I want you to go ahead and DO IT. It is much easier to apologize than it is to get permission. – Grace Hopper

I think a lot of the basis of the open source movement comes from procrastinating students. – Andrew Tridgell







Life would be much easier if I had the source code. – Anonymous

Thank you for attending!

References I

-  Android HCE DESFire: A software implementation of Desfire in an Android app. <https://github.com/jekkos/android-hce-desfire>
-  Chameleon Mini Firmware (authoritative sources). <https://github.com/emsec/ChameleonMini>
-  ISO/IEC 14443, 15693 and 7816 Standards. Identification Cards - Contactless Integrated Circuit Cards. www.iso.org
-  Kasper T., von Maurich I., Oswald D., Paar C. (2011) Chameleon: A Versatile Emulator for Contactless Smartcards. In: Rhee KH., Nyang D. (eds) Information Security and Cryptology - ICISC 2010. ICISC 2010. Lecture Notes in Computer Science, vol 6829. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24209-0_13
-  Kasper, T. and Oswald, D. Presentation slides on the history of the Chameleon Mini devices. https://raw.githubusercontent.com/wiki/emsec/ChameleonMini/Images/160110_ChameleonMini_history_smaller.pdf

References II

-  LibFreeFare: A convenience API for NFC cards manipulations on top of LibNFC. <https://github.com/nfc-tools/libfreefare>
-  LibNFC: A platform independent NFC library. <https://github.com/nfc-tools/libnfc>
-  Microchip. ATxmega1284U Data Sheet. <https://ww1.microchip.com/downloads/en/DeviceDoc/ATxmega128-64-32-16A4U-DataSheet-DS40002166A.pdf>
-  NXP Semiconductors. MIFARE DESFire Functional specification. Publicly available MF3ICD81 datasheet (2008). <https://tinyurl.com/kwweanp9>
-  Philips Semiconductors. Mifare DESFire: Contactless multi-application IC with DES and 3DES security. Publicly available MF3-IC-D40 datasheet (2004). <https://tinyurl.com/5era3dx2>
-  Proxmark III. A Radio Frequency IDentification Tool. <http://www.proxmark.org>

References III



Schmidt, M. D. Chameleon Mini DESFire Stack (development sources).
<https://github.com/maxieds/ChameleonMiniDESFireStack>



Schmidt, M. D. Chameleon Mini Live Debugger.
<https://github.com/maxieds/ChameleonMiniLiveDebugger>