MIFARE SDK

Public

MobileKnowledge
June 2015

Agenda

- Overview of MIFARE SDK related technologies
    - NFC Technology (Read/Write mode)
    - MIFARE, NTAG and ICODE products
- NFC in Android
- MIFARE SDK
    - Introduction to the MIFARE SDK library
    - How to start using the library
    - MIFARE SDK Lite Edition vs Advanced Edition
- MIFARE SDK code examples
- Use Cases

NFC Technology

Read/Write mode

Card Emulation

Peer to Peer

Read/Write

Reads / Writes data from any tag or contactless card

MIFARE SDK

NXP Products

MIFARE

Broadest product portfolio tailored to more than 40 different applications

Broadest product portfolio tailored to the automatic fare collection market

Leading product families are MIFARE Classic, MIFARE Ultralight, MIFARE Plus, MIFARE DESFire and SmartMX

ntag

Ideal choice for mass market deployment of NFC proximity marketing and electronics pairing applications

Combines ease of integration, high RF sensitivity and anti-cloning features

NTAG I2C connected tag integrates a I2C contact interface in addition to the passive NFC Forum compliant interface

ICODE

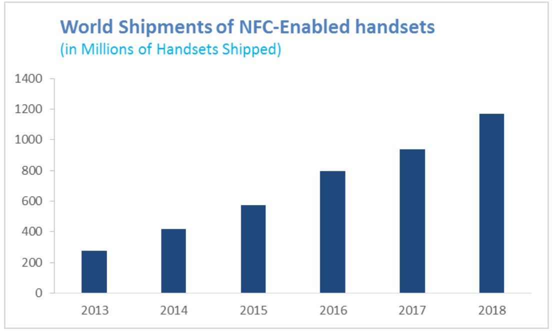
Industry standard for high-frequency (HF) smart label solutions. Broadest product portfolio tailored to the automatic fare collection market

Billions of ICs in the field and thousands of successful installations

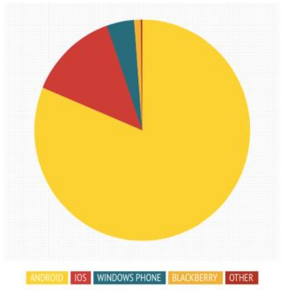
NFC in Android

Android NFC Market Update

- Global Smartphone sales exceeded 1.2 Billion units in 2014. 20% year-on-year increase registered.
- Smartphones share expected to continue growing from 67% in 2014 to > 80% or even higher in the coming years
- 3 in 4 mobile phones to come with NFC by 2018



- All major OEMs supporting Android integrate NFC technology
- Android accounts for more than 75% of Mobile OS market share
  - +1.5M apps on the Play Store
  - +450K Publishers
  - +1.5B downloads from the Play Store every month
  - +1M devices activated worldwide everyday



Global Smartphone Shipments

NFC in Android

- Read/Write mode supported
  - Passive NFC Forum Tags
    - Tag Type 1: Topaz
    - Tag Type 2: MIFARE Ultralight & NTAG (simple dedicated API)
    - Tag Type 3: FeliCa
    - Tag Type 4: MIFARE DESFire
  - Proprietary NXP NFC Tags
    - MIFARE Classic (simple dedicated API)
    - ICODE
- Peer to Peer mode supported
- Card Emulation mode “supported”
  - HCE supported since Android KitKat

- Android NFC developer’s guide
  - http://developer.android.com/guide/topics/connectivity/nfc/index.html

NFC in Android

My first MIFARE DESFire-based application

- Connect to the card and exchange data
  - Class to use: android.nfc.tech.IsoDep class ??
  - Commands to be exchanged in hexadecimal !!
- Advanced technical knowledge needed
  - MIFARE DESFire EV1 datasheet …
  - ISO 7816-4 specification …
  - ISO/IEC 14443 standard …
- Manage the MIFARE DESFire AES-based cryptography
  - CMAC calculator
  - CRC32 calculator
  - Initialization Vector management

```
→ 90 0a 00 00 01 00 00
← a2 de cd 02 c8 46 2b 31 95 af
→ 90 af 00 00 10 b0 cc bc ed 4f c8 32 c9 08 dc e2 4d 86 ca ec 3c 00
← 76 73 d9 49 71 3f f2 d1 91 00
```

- Users care about the User Interface and application interaction
  - The time you invest managing the contactless communication, the time you do not invest developing your cool app

MIFARE SDK

MIFARE SDK
Introduction

- Extensive software development tool that lets developers create contactless applications for the complete portfolio of MIFARE, NTAG and ICODE products on any NFC-enabled devices.
- Software and Hardware KeyStore supporting NXP's SAM AV2 module for the development of secure apps.
- Complete product support package: user manual, documentation, examples, …

http://www.mifare.net/en/home/
http://www.mifare.net/en/products/mifare-sdk/

MIFARE SDK

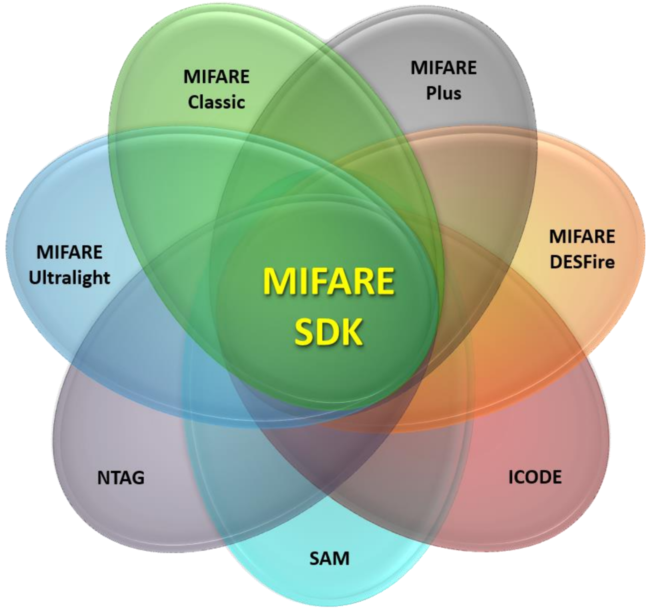
Why should I use it?

- MIFARE SDK is ideal for building reliable, interoperable and scalable applications for smartphones
- Developers are able to benefit from an enormous reduction in development time.
  - Developers focus on designing creative apps and the best GUI for their brands.
  - Short time from idea to market
- Get rid of “complicated” datasheets and application notes
  - Full command set support on Java level
- Leverage the worldwide success of NXP’s product installations.
- Comprehensive documentation with User Manual and Javadoc documentation
- Source code examples to get familiar with the technology as fast as possible
- Talk to our experts on the MIFARE SDK Forum

MIFARE SDK Content

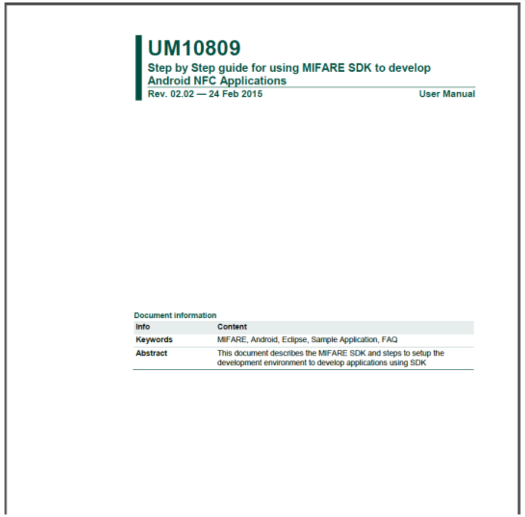
- The MIFARE SDK package contains:
  - Java library file (to import in your programming IDE)
  - Complete Javadoc documentation with the API description
  - User Manual describing how to start and use the SDK
  - Sample reference applications
  - Release note

- Requirements:
  - Software
    - Android Development Tool environment from Google
    - [HIC Omnikey Driver for Android]
  - Hardware
    - Android NFC device with Android 4.x (ICS) and above
    - [HID SAM reader]



MIFARE SDK Documentation

User Manual and Javadoc documentation



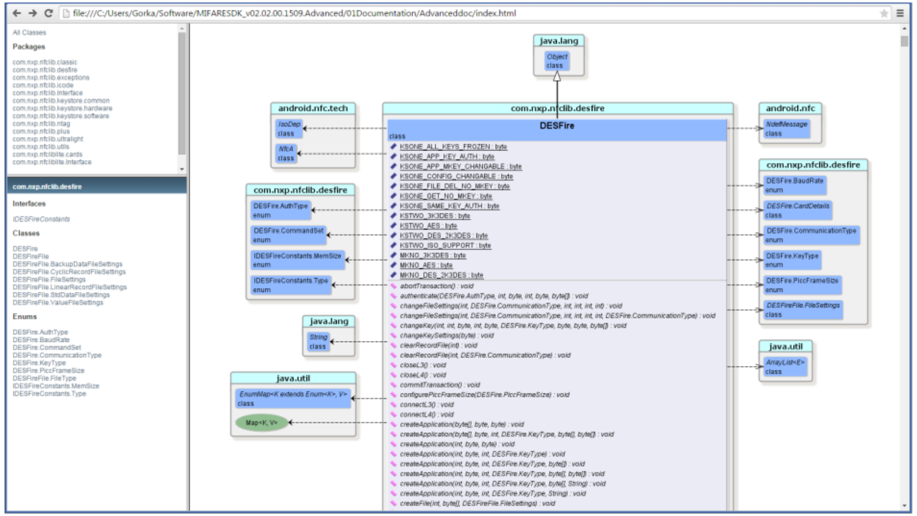
UM10809

Step by Step guide for using MIFARE SDK to develop Android NFC Applications

Rev. 02.02 — 24 Feb 2015 **User Manual**

Document information

Info	Content
Keywords	MIFARE, Android, Eclipse, Sample Application, FAQ
Abstract	This document describes the MIFARE SDK and steps to setup the development environment to develop applications using SDK



file:///C:/Users/Gorka/Software/MIFARESDK_v02.02.00.1509.Advanced/01Documentation/Advanceddoc/index.html

All Classes

Packages

com.nxp.nfclib.classic
com.nxp.nfclib.desfire
com.nxp.nfclib.exceptions
com.nxp.nfclib.icode
com.nxp.nfclib.interface
com.nxp.nfclib.keystore.common
com.nxp.nfclib.keystore.hardware
com.nxp.nfclib.keystore.software
com.nxp.nfclib.ntag
com.nxp.nfclib.plus
com.nxp.nfclib.ultralight
com.nxp.nfclib.utils
com.nxp.nfcliblite.cards
com.nxp.nfcliblite.interface

com.nxp.nfclib.desfire

Interfaces

IDESFireConstants

Classes

DESFire
DESFireFile
DESFireFile.BackupDataFileSettings
DESFireFile.CyclicRecordFileSettings
DESFireFile.FileSettings
DESFireFile.LinearRecordFileSettings
DESFireFile.StdDataFileSettings
DESFireFile.ValueFileSettings

Enums

DESFire.AuthType
DESFire.BaudRate
DESFire.CommandSet
DESFire.CommunicationType
DESFire.KeyType
DESFire.PiccFrameSize
DESFireFile.FileType
IDESFireConstants.MemSize
IDESFireConstants.Type

```
com.nxp.nfclib.desfire
DESFire
class
KSONE_ALL_KEYS_FROZEN : byte
KSONE_APP_KEY_AUTH : byte
KSONE_APP_MKEY_CHANGABLE : byte
KSONE_CONFIG_CHANGABLE : byte
KSONE_FILE_DEL_NO_MKEY : byte
KSONE_GET_NO_MKEY : byte
KSONE_SAME_KEY_AUTH : byte
KSTWO_3K3DES : byte
KSTWO_AES : byte
KSTWO_DES_2K3DES : byte
KSTWO_ISO_SUPPORT : byte
MKNO_3K3DES : byte
MKNO_AES : byte
MKNO_DES_2K3DES : byte
abortTransaction() : void
authenticate(DESFire.AuthType, int, byte, int, byte, byte[]) : void
changeFileSettings(int, DESFire.CommunicationType, int, int, int, int) : void
changeFileSettings(int, DESFire.CommunicationType, int, int, int, int, DESFire.CommunicationType) : void
changeKey(int, int, byte, int, byte, DESFire.KeyType, byte, byte, byte[]) : void
changeKeySettings(byte) : void
clearRecordFile(int) : void
clearRecordFile(int, DESFire.CommunicationType) : void
closeL3() : void
closeL4() : void
commitTransaction() : void
configurePiccFrameSize(DESFire.PiccFrameSize) : void
connectL3() : void
connectL4() : void
createApplication(byte[], byte, byte) : void
createApplication(byte[], byte, int, DESFire.KeyType, byte[], byte[]) : void
createApplication(int, byte, byte) : void
createApplication(int, byte, int, DESFire.KeyType) : void
createApplication(int, byte, int, DESFire.KeyType, byte[]) : void
createApplication(int, byte, int, DESFire.KeyType, byte[], byte[]) : void
createApplication(int, byte, int, DESFire.KeyType, byte[], String) : void
createApplication(int, byte, int, DESFire.KeyType, String) : void
createFile(int, byte[], DESFireFile.FileSettings) : void
```

MIFARE SDK User Manual

Introduction to the MIFARE SDK and explanation on how to integrate the MIFARE SDK in your project and start developing

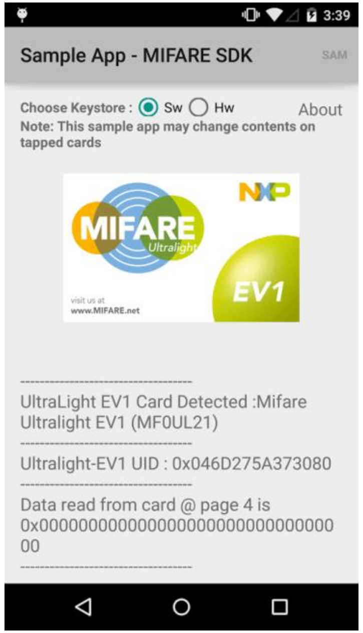
Javadoc documentation

Complete API description ideal for programmers

Javadoc documentation can be consulted as an interactive website and integrated into the development IDE for further consulting during coding phase

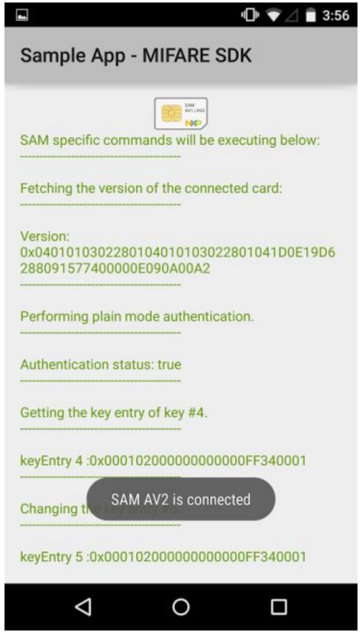
MIFARE SDK Sample App

- Sample App downloadable from the Play Store
- Application that detects any card and demonstrates read/write of data onto the card
  - It supports MIFARE, NTAG and ICODE products
- Hardware KeyStore is demonstrated using HID OMNIKEY readers with NXP's SAM inserted into it
- Source code available in the MIFARE SDK package



Choose Keystore : [✓ Sw] Hw
Note: This sample app may change contents on tapped cards

```
--------------------------------
UltraLight EV1 Card Detected :Mifare
Ultralight EV1 (MF0UL21)
--------------------------------
Ultralight-EV1 UID : 0x046D275A373080
--------------------------------
Data read from card @ page 4 is
0x00000000000000000000000000000000
00
--------------------------------
```

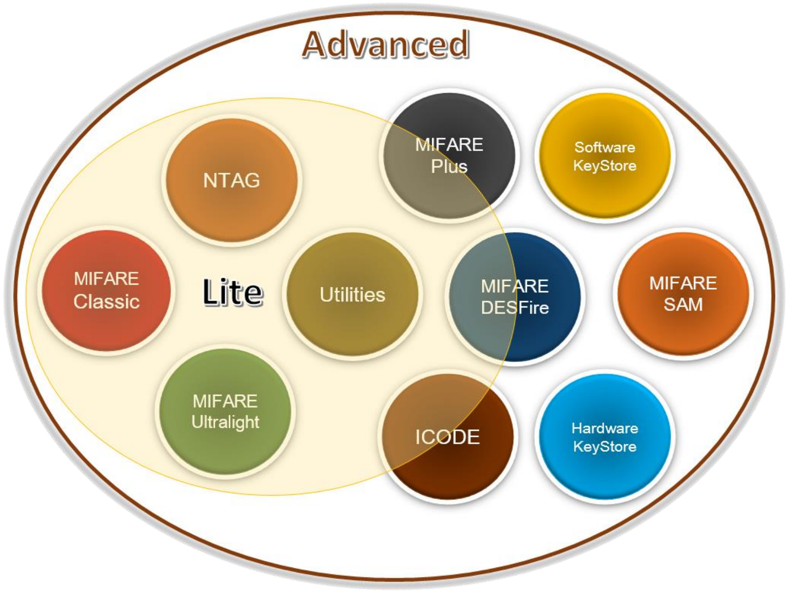


```
SAM specific commands will be executing below:
--------------------------------
Fetching the version of the connected card:
--------------------------------
Version:
0x040101030228010401010302280104 1D0E19D6
288091577400000E090A00A2
--------------------------------
Performing plain mode authentication.
--------------------------------
Authentication status: true
--------------------------------
Getting the key entry of key #4.
--------------------------------
keyEntry 4 :0x0001020000000000000FF340001
--------------------------------
Changing t
--------------------------------
keyEntry 5 :0x0001020000000000000FF340001
```

MIFARE SDK
Lite vs Advanced version

LITE version offers a reduced API for simple use cases such as read/write operations and single NDEF operations

Advanced version offers a complete API for all MIFARE cards and supports all type of operations. Software and Hardware KeyStore are only supported in this version.



MIFARE SDK Lite version

Getting started

STEP 1

STEP 2

STEP 3

Login & Download

Login in the MIFARE SDK website and download MIFARE SDK Lite version for free

Install

Follow the MIFARE SDK User Manual in order to integrate the java library in your Android project

Code

Start developing cool NFC apps that leverage on MIFARE, NTAG and ICODE infrastructure

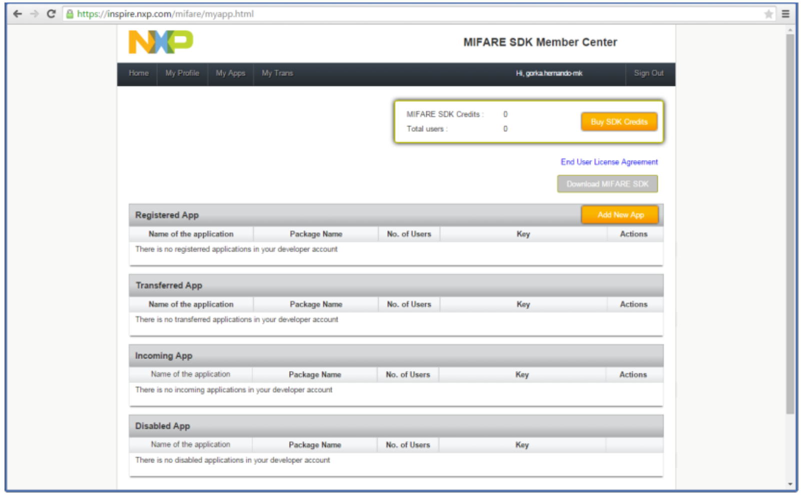
Training Mobile Knowledge

MIFARE SDK Advanced version

Licensing and getting started

Login → **Buy credits** → **Download SDK** → **Register your app** → **Start coding**

Login	Buy credits	Download SDK	Register your app	Start coding
Login in the MIFARE SDK website	Purchase credits: 1000 Credits: 99€ 5000 Credits: 399€	Download the MIFARE SDK Advanced version from the website	Register the app where the MIFARE SDK will be used	Use the obtained key in your app and start coding cool NFC apps


https://inspire.nxp.com/mifare/myapp.html

MIFARE SDK Member Center

Home | My Profile | My Apps | My Trans | Hi, gorka.hernando-mk | Sign Out

MIFARE SDK Credits : 0
Total users : 0

Buy SDK Credits

End User License Agreement

Download MIFARE SDK

Registered App — Add New App

Name of the application	Package Name	No. of Users	Key	Actions
There is no registerred applications in your developer account				

Transferred App

Name of the application	Package Name	No. of Users	Key	Actions
There is no transferred applications in your developer account				

Incoming App

Name of the application	Package Name	No. of Users	Key	Actions
There is no incoming applications in your developer account				

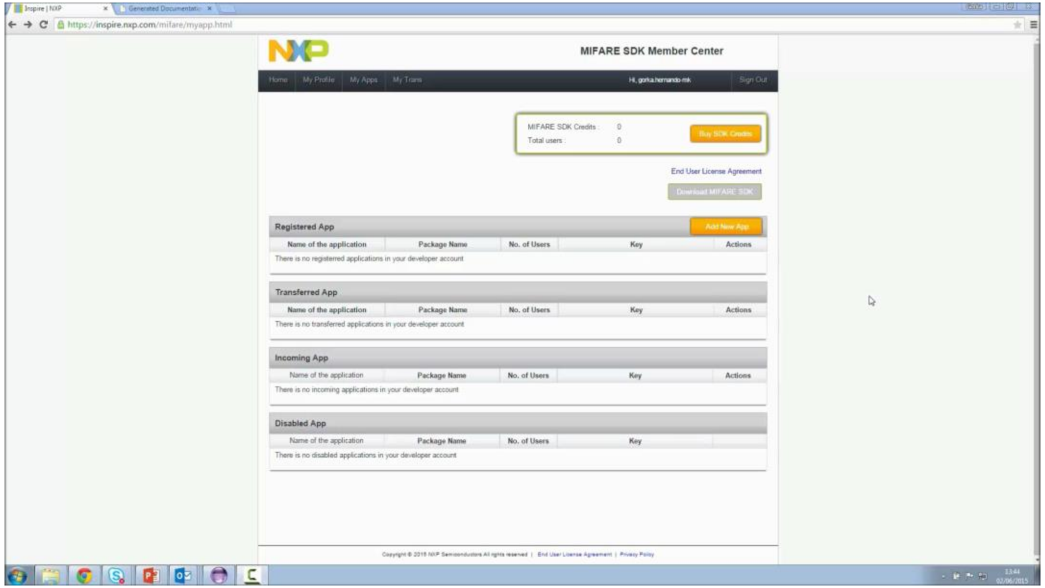
Disabled App

Name of the application	Package Name	No. of Users	Key	
There is no disabled applications in your developer account				

Training Mobile Knowledge

Practical exercise

How to start building your MIFARE SDK apps



MIFARE SDK Member Center

MIFARE SDK Credits : 0

Total users : 0

Registered App

Name of the application	Package Name	No. of Users	Key	Actions
There is no registered applications in your developer account				

Transferred App

Name of the application	Package Name	No. of Users	Key	Actions
There is no transferred applications in your developer account				

Incoming App

Name of the application	Package Name	No. of Users	Key	Actions
There is no incoming applications in your developer account				

Disabled App

Name of the application	Package Name	No. of Users	Key	
There is no disabled applications in your developer account				

Copyright © 2015 NXP Semiconductors All rights reserved | End User License Agreement | Privacy Policy

https://youtu.be/AsDZT101Zrk

MIFARE SDK

My first MIFARE DESFire-based application

- Dedicated DESFire class available
  - No hexadecimal commands to be sent
- High-level Java API for operating on the card
  - Authenticate
  - Read
  - Write
  - ChangeKey
  - …
- Advanced technical knowledge not needed anymore
- Manage the MIFARE DESFire AES-based cryptography
  - The MIFARE SDK will manage it for you
    - Software and Hardware KeyStore
- Developers invest the majority of their time in the application logic and User Interface

```
objDESFire.connect();
objDESFire.authenticate(AppId, deskey);
objDESFire.write (data);
```

MIFARE SDK

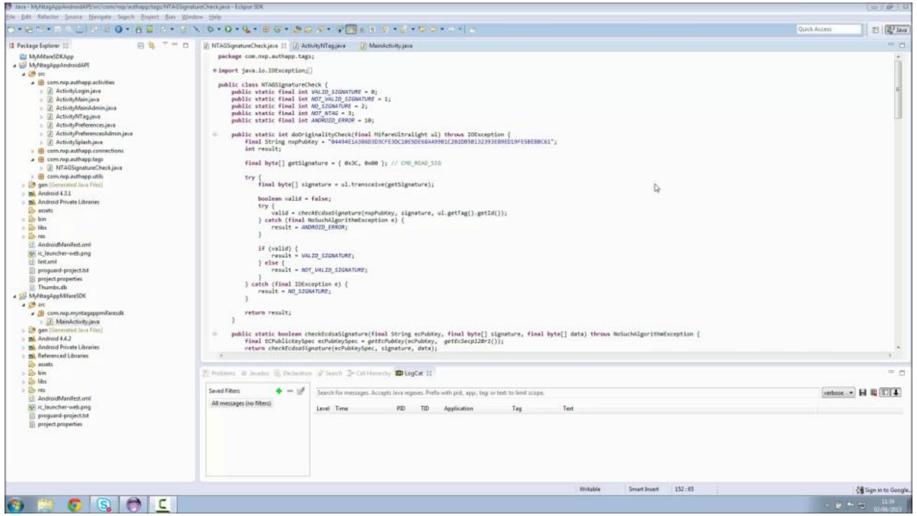
New features and updates

- Latest features in Advanced Version v02.02 and v02.01:
  - Root check removed
  - ICODE SLIX2 support added
  - PlusSL1 class is added for detecting Security Level 1 separately
  - GetCardDetails API is made uniform across cards
  - Added MakeReadOnly API for MIFARE Ultralight and NTAG
  - Fixed Ultralight C CounterIncrement API
  - …
- New features to come
  - Full MIFARE DESFire EV2 command set support
  - Other SAM form-factors
  - New platforms support
  - Utilities, tools, APIs, …

MIFARE SDK Sample code

Practical exercise

MIFARE SDK Sample Code I

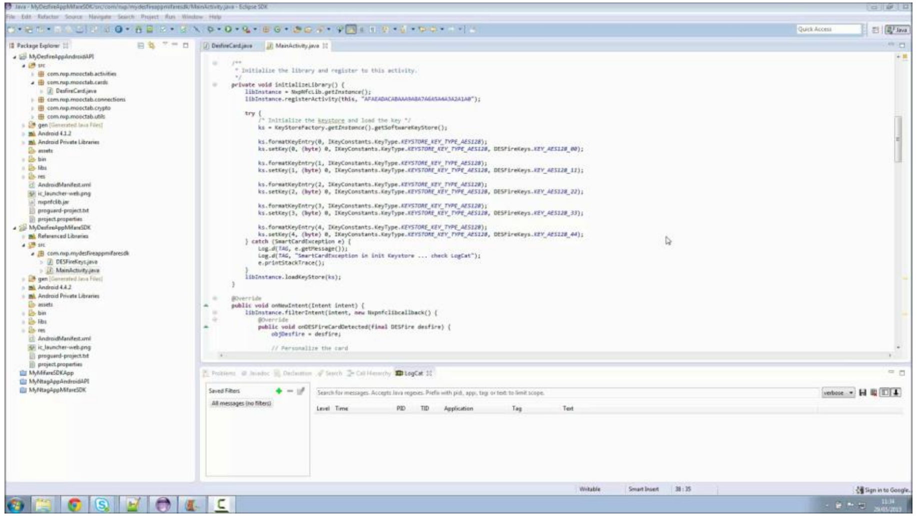


```
package com.nxp.authapp.tags;

+import java.io.IOException;

public class NTAGSignatureCheck {
    public static final int VALID_SIGNATURE = 0;
    public static final int NOT_VALID_SIGNATURE = 1;
    public static final int NO_SIGNATURE = 2;
    public static final int NOT_NTAG = 3;
    public static final int ANDROID_ERROR = 10;

    public static int doOriginalityCheck(final MifareUltralight ul) throws IOException {
        final String nxpPubKey = "04494E1A386D3D3CFE3DC10E5DE68A499B1C202DB5B132393E89ED19FE5BE8BC61";
        int result;

        final byte[] getSignature = { 0x3C, 0x00 }; // CMD_READ_SIG

        try {
            final byte[] signature = ul.transceive(getSignature);

            boolean valid = false;
            try {
                valid = checkEcdsaSignature(nxpPubKey, signature, ul.getTag().getId());
            } catch (final NoSuchAlgorithmException e) {
                result = ANDROID_ERROR;
            }

            if (valid) {
                result = VALID_SIGNATURE;
            } else {
                result = NOT_VALID_SIGNATURE;
            }
        } catch (final IOException e) {
            result = NO_SIGNATURE;
        }

        return result;
    }

    public static boolean checkEcdsaSignature(final String ecPubKey, final byte[] signature, final byte[] data) throws NoSuchAlgorithmException {
        final ECPublicKeySpec ecPubKeySpec = getEcPubKey(ecPubKey,  getEcSecp128r1());
        return checkEcdsaSignature(ecPubKeySpec, signature, data);
```

https://youtu.be/GAO1KMs646c

Practical exercise

MIFARE SDK Sample Code II

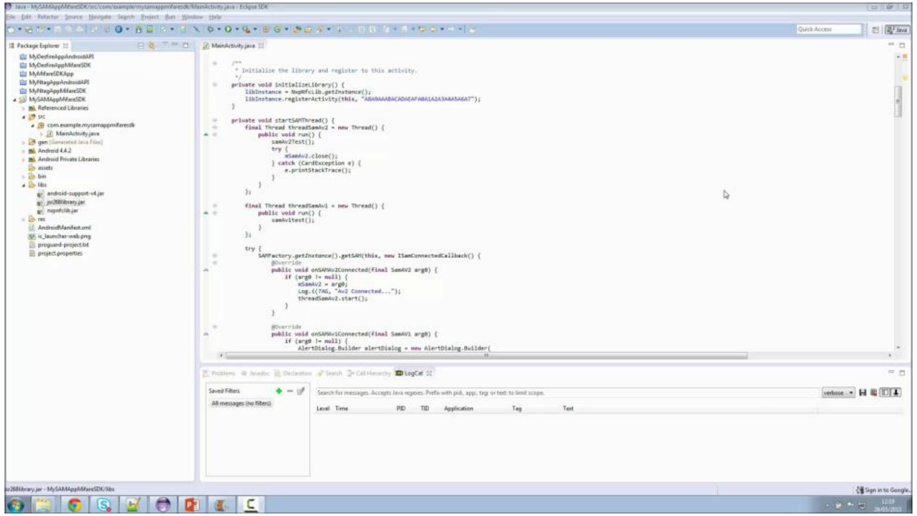


```java
/**
 * Initialize the library and register to this activity.
 */
private void initializeLibrary() {
    libInstance = NxpNfcLib.getInstance();
    libInstance.registerActivity(this, "AFAEADACABAAA9A8A7A6A5A4A3A2A1A0");

    try {
        /* Initialize the keystore and load the key */
        ks = KeyStoreFactory.getInstance().getSoftwareKeyStore();

        ks.formatKeyEntry(0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128);
        ks.setKey(0, (byte) 0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128, DESFireKeys.KEY_AES128_00);

        ks.formatKeyEntry(1, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128);
        ks.setKey(1, (byte) 0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128, DESFireKeys.KEY_AES128_11);

        ks.formatKeyEntry(2, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128);
        ks.setKey(2, (byte) 0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128, DESFireKeys.KEY_AES128_22);

        ks.formatKeyEntry(3, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128);
        ks.setKey(3, (byte) 0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128, DESFireKeys.KEY_AES128_33);

        ks.formatKeyEntry(4, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128);
        ks.setKey(4, (byte) 0, IKeyConstants.KeyType.KEYSTORE_KEY_TYPE_AES128, DESFireKeys.KEY_AES128_44);
    } catch (SmartCardException e) {
        Log.d(TAG, e.getMessage());
        Log.d(TAG, "SmartCardException in init Keystore ... check LogCat");
        e.printStackTrace();
    }
    libInstance.loadKeyStore(ks);
}

@Override
public void onNewIntent(Intent intent) {
    libInstance.filterIntent(intent, new Nxpnfclibcallback() {
        @Override
        public void onDESFireCardDetected(final DESFire desfire) {
            objDesfire = desfire;

            // Personalize the card
```

https://youtu.be/EjVdlpg5OG8

Practical exercise

MIFARE SDK Sample Code III



```
    /**
     * Initialize the library and register to this activity.
     */
    private void initializeLibrary() {
        libInstance = NxpNfcLib.getInstance();
        libInstance.registerActivity(this, "ABA0AAABACADAEAFABA1A2A3A4A5A6A7");
    }

    private void startSAMThread() {
        final Thread threadSamAv2 = new Thread() {
            public void run() {
                samAv2Test();
                try {
                    mSamAv2.close();
                } catch (CardException e) {
                    e.printStackTrace();
                }
            }
        };

        final Thread threadSamAv1 = new Thread() {
            public void run() {
                samAv1test();
            }
        };

        try {
            SAMFactory.getInstance().getSAM(this, new ISamConnectedCallback() {
                @Override
                public void onSAMAv2Connected(final SamAV2 arg0) {
                    if (arg0 != null) {
                        mSamAv2 = arg0;
                        Log.i(TAG, "Av2 Connected...");
                        threadSamAv2.start();
                    }
                }

                @Override
                public void onSAMAv1Connected(final SamAV1 arg0) {
                    if (arg0 != null) {
                        AlertDialog.Builder alertDialog = new AlertDialog.Builder(
```

https://youtu.be/HS2P0cix8_Q

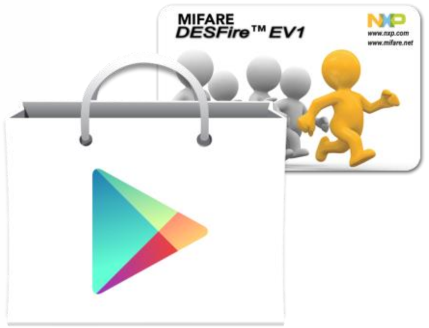
Use Cases

MIFARE SDK
Where to use it

- Smartcard-enabled Android applications
- Access management
- Closed-loop micropayment
- Campus and student cards
- Loyalty programs, couponing and gift card applications
- Gaming
- Libraries
- Smart homes
- Consumer interaction
- Smart media
- …

MIFARE SDK

Loyalty Use case



Idea

My restaurant application with menus, reservations, … in the Play Store.

MIFARE-based Loyalty card service as the way to succeed

Development

Develop application using Android API, MIFARE SDK and cloud services

Application logic: 4 hours
Application GUI: 2 hours
MIFARE logic: 15 minutes

Publish

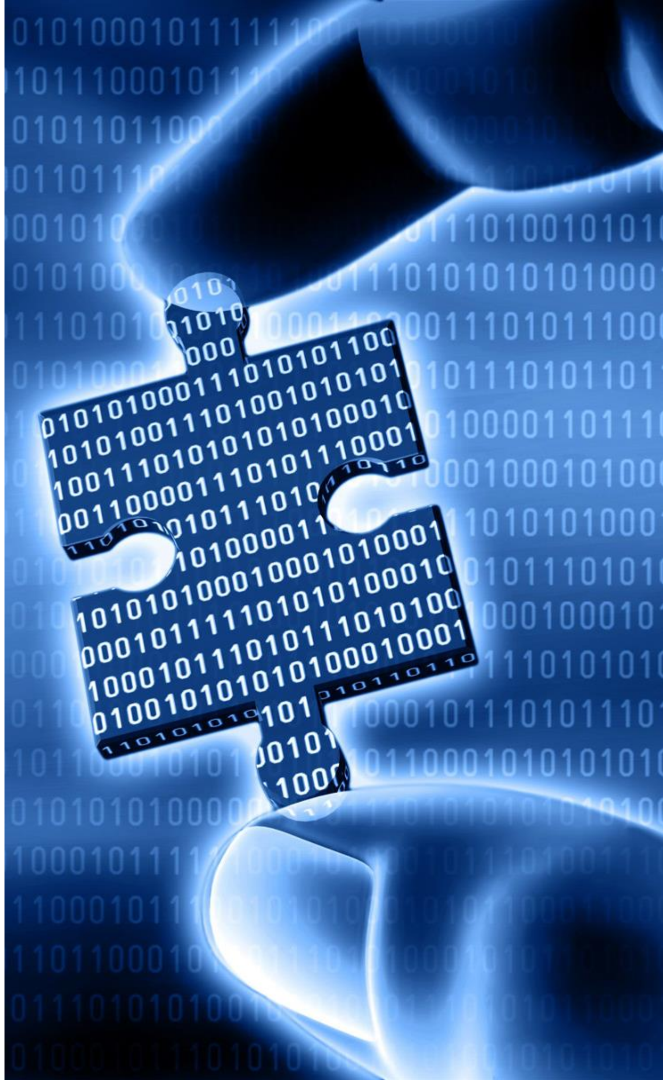
Publish application in the Play Store and wait for new customers thanks to my brand new MIFARE-based Loyalty program!!!

Conclusion

MIFARE SDK

Wrap up

- Smartphone applications are a great business opportunity
  - Make your application stand out with NFC technology
- Managing contactless communication is not easy using Android API
- MIFARE SDK helps you to develop reliable, interoperable and scalable applications that rely on NXP products
  - High-level Java API for contactless communication
  - Complete and comprehensive documentation
  - Source code examples
  - Support to developers
  - Integration of new products guaranteed



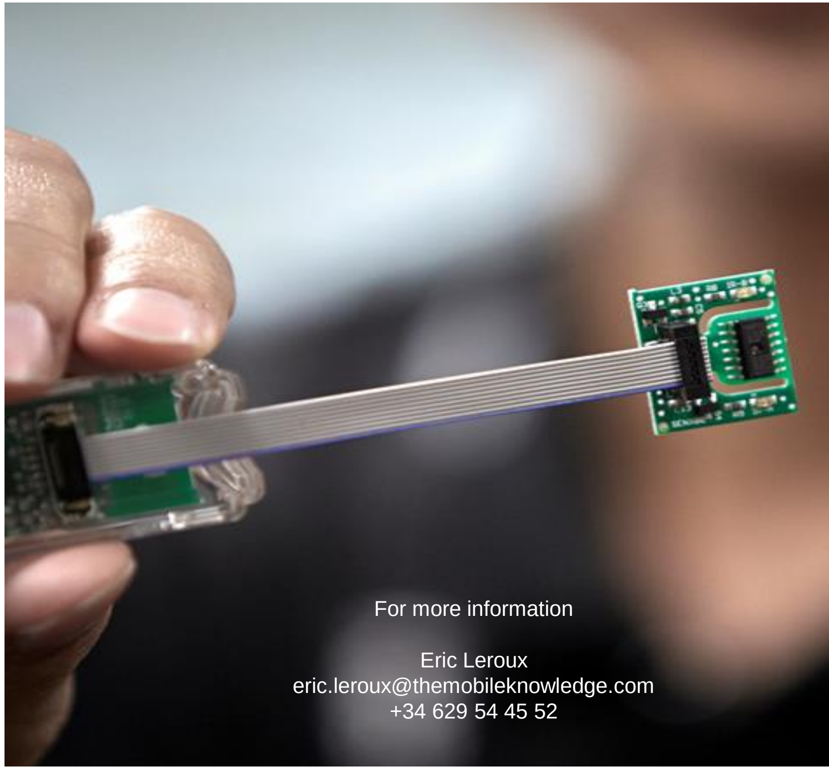
Training Mobile Knowledge

MobileKnowledge

Thank you for your attention

www.themobileknowledge.com

- We are a global competence team of hardware and software technical experts in all areas related to contactless technologies and applications.
- Our services include:
  - Application and system Design Engineering support
  - Project Management
  - Technological Consulting
  - Advanced Technical Training services
- We address all the exploding identification technologies that include NFC, secure micro-controllers for smart cards and mobile applications, reader ICs, smart tags and labels, MIFARE family and authentication devices.



For more information

Eric Leroux
eric.leroux@themobileknowledge.com
+34 629 54 45 52

Training