InteropEHRate

D3.12

Libraries for remote and D2D IDM in mobile and web applications - V1

ABSTRACT

This deliverable describes a demonstration of the initial version of the security libraries offered by the InteropEHRate Framework as a reference implementation of the identity management (IDM) in mobile and web applications for health record exchange. It also outlines for these libraries their description regarding their current version and the used licences, as well as specific information regarding their development, followed by specific guidelines regarding the installation and the usage of these libraries, through a detailed guide. This software will be compliant to the specification released by project Task "*Identity management and encryption mechanisms*".

Delivery Date	January 10th, 2020
Work Package	WP3
Task	T3.4
Dissemination Level	Public
Type of Deliverable	Demonstrator
Lead partner	UBITECH LIMITED [UBIT]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

CONTRIBUTORS

	Name	Partner
Contributors	Sofianna Menesidou, Dimitris Papamartzivanos	UBIT

LOGTABLE

Version	Date	Change	Author	Partner
0.1	05-12-19	First draft of ToC	D. Papamartzivanos, S. Menesidou	UBIT
0.2	10-12-19	Introduction	D. Papamartzivanos	UBIT
0.3	13-12-19	SW Description	S. Menesidou	UBIT
0.4	17-12-19	SW Description	S. Menesidou	UBIT
0.5	18-12-19	SW Description	S. Menesidou	UBIT
0.6	27-12-19	Overview	S. Menesidou	UBIT
0.7	02-01-20	Overview	S. Menesidou	UBIT
0.8	08-01-20	Structure	D. Papamartzivanos	UBIT
0.9	09-01-20	Quality Review	Argyro Mavrogiorgou	UPRC
vFinal	10-01-20	Final review and version for submission	Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
D2D	Device-to-Device
HCP	Healthcare Professional
HR	Health Record
IDM	Identity Management
S-EHR	Smart Electronic Health Record
SW	Software

TABLE OF CONTENT

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of the document

The main goal of this document is to deliver a demonstration of the initial version (V1) of the security libraries offered by the InteropEHRate Framework as a reference implementation of the secure health record exchange. In more detail, the current document outlines for these libraries their description regarding their current version and the used licences, as well as specific information regarding their development (e.g. programming languages used, supported platforms, used libraries, versions etc). Moreover, this document includes specific guidelines regarding the installation and the usage of these libraries, through a detailed guide.

It should be noted that this document is a SW report, a pointer to the actual deliverable regarding the design of the libraries for HR security and privacy services **[D3.9]**. In addition, the security libraries are closely related and dependent to the relying exchange protocol and for that reason, the same code base provided in **[D.12]** is extended to support the necessary security features for identity management (IDM).

1.2. Intended audience

The document is intended to security engineers, developers, architects, and all the InteropEHRate project participants and partners interested to have an overview of how InteropEHRate will support HR security and privacy services. Apart from that, the document is intended for researchers and developers as well, as they may be interested in installing and using the designed security libraries.

1.3. Structure of the document

The current document is organized in the following Sections:

- **Section 1** introduces the overall concept of the document, defining its scope, intended audience, and relation to the other project tasks and reports.
- **Section 2** outlines the description of the software (SW) regarding the offered security libraries, including details such as their programming languages, and their supported platforms.
- **Section 3** describes for each library both an installation and a user guide.

1.4. Updates with respect to previous version (if any)

Not applicable, since this document is the initial version of the demonstration of the security libraries for the identity management (IDM).

2. SW DESCRIPTION

2.1.S-EHR side Security Library

SW TITLE	Mobile D2D Security Management (M-D2D-SM)
SW VERSION	1.0
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Java SE 8.0
SUPPORTED PLATFORM(s)	Android (5.0 - Current Version)
SOURCE CODE	http://iehrgitlab.ds.unipi.gr/interopehrate/s-ehr-mobile-app/d2d-hr-exchange/tree/security
EXECUTABLE	N.A.

Table 1 – S-EHR side security library description

2.2.HCP side Security Library

SW TITLE	Terminal D2D Security Management (T-D2D-SM)
SW VERSION	1.0
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Java SE 8.0
SUPPORTED PLATFORM(s)	Windows OS
SOURCE CODE	http://iehrgitlab.ds.unipi.gr/interopehrate/reference-hcp-app/terminal-d2d-hr-exchange/tree/security
EXECUTABLE	N.A.

Table 2 – HCP side security library description

3. OVERVIEW

3.1.S-HER side Security Library

The current release of the mobile security library (i.e. M-D2D-SM), contains all the operations of the 1st variant that are needed from the side of the S-EHR application developer to interact with the D2D library and the reference S-EHR application. This library contains different operations that have to be invoked for implementing the security functions of the D2D protocol in the context of the mobile application. This library is a Java-based component that can be nested in any Android application. It offers a set of Java operations for identity management. More details regarding these operations can be found in deliverables **[D3.3]** and **[D3.9]**.

3.1.1. Installation guide

The installation guide of the security D2D library inside the S-EHR app contains the necessary information for the process that has to be followed in order to add the library to the S-EHR app Android project, and make sure that every component of the S-EHR app is able to use the security library. As already mentioned, the security library extends the same code base of the D2D protocol due to some necessary dependencies. Hence, the installation guide introduced in **[D4.12]** is the same for the security library. For completeness purposes we provided it also in this deliverable. This guide is used to install the version of the library released in December 2019, and may be no longer valid for more recent versions. The only requirement for this process is for the developer to have an Android project application running with the min-SDK version properties upper or equal to 15.

Installation steps of the S-EHR side security D2D library:

1. Insert into the build.gradle file of the project the following lines of code inside the repository section to retrieve libraries:

```
repositories {
    google()
    jcenter()
    maven {
   url 'http://213.249.46.206:8081/repository/maven-releases/'
   content {
     includeGroup 'eu.interoperhate'
     }
   }
}
```

2. Insert into the build.gradle file of the module the following dependency:

```
implementation(group:'eu.interoperhate',name:'md2de',version: '0.0.1')
```

3. Refresh the gradle project.

3.1.2. User guide

This user guide refers to the version 1.0 (V1) of the IDM functionalities of the S-EHR side as an extension of the D2D library. The main flow of the functionality of this library is the following:

- The `fetchCertificate()` function will be called to acquire from the CA the necessary Certificates and store them to the `Android Keystore`.

```
public KeyStore fetchCertificate()

throws NoSuchAlgorithmException, NoSuchProviderException,
CertificateException, OperatorCreationException,
InvalidKeyException, SignatureException;
```

- In addition to the Bluetooth MAC address of the HCP app running machine, the signed MAC address will also be acquired for later identity verification. The signature will be stored to the `SharedPreferences` of the S-EHR app to be checked after the successful connection and certificate transmission from the HCP.

```
public void storeScanned(Context context, String
scannedSignature, String scannedAddress);
```

- After the successful connection, HCP's identity in the form of Certificate is received from the HCP web app. The Certificate is signed by Health Organization's Certificate and Health Organization's Certificate is signed by InteropEHRate trusted CA. To handle the Certificate reception, we need the `onCertRecivedStore()` function that stores the Certificate to the Android KeyStore. This function returns true upon successful Certificate storage and false in case of failure.

```
public boolean onCertRecivedStore(Certificate cert)
```

- After the successful reception of the Certificate, the S-EHR app acquires the stored signature, the Bluetooth MAC address and received Certificate (in the form of RSA Public Key). Finally, it checks the validity of the signature and proves the signer's identity.

```
public boolean verifySignature(RSAPublicKey publicKey, byte[]
scannedAddress, byte[] scannedSignature)
throws UnsupportedEncodingException, NoSuchAlgorithmException,
InvalidKeyException, SignatureException;
```

- The `sendSEHRCertificate()` function is called immediately after the connection is established and sends to the HCP App the following Base64 representation of the Certificate.

```
public void sendSEHRCertificate(KeyStore keyStore)

throws IOException, KeyStoreException, CertificateException,
NoSuchAlgorithmException;
```

3.2. HCP side Security Library

The current release of the library contains all the operations that are needed from the side of the HCP application developer to interact with the D2D library and the reference HCP application. This library contains different operations that have to be invoked in a specific sequence for implementing the security functions of the D2D protocol. This library is a Java based component that can be embedded in any Java-based application. It offers a set of Java operations for identity management. More details regarding these operations can be found in deliverables **[D3.3]** and **[D3.9]**.

3.2.1. Installation guide

The installation guide of the security D2D library inside the HCP contains necessary information for the process that has to be followed in order to add the library to the HCP Java project and make sure that every component of the HCP is able to use the security library. As already mentioned, the security library extends the same code base of the D2D protocol due to some necessary dependencies. Hence the installation guide introduced in **[D4.12]** is the same for the security library. For the sake of completeness, we provided it also in this deliverable. This guide is used to install the version of the library released in December 2019 and may be no longer valid for more recent versions. The only requirement for this process is for the developer to have Git and a Java 8 JDK installed. This guide is used to install the version of the library released in December 2019, and may be no longer valid for more recent versions.

Installation steps of the HCP side security D2D library:

1. Access the library's Java Project repository that is located on GitLab EHR private repository and clone the Terminal D2D HR Exchange.
2. Change to the security branch of the HCP app.
   http://iehrgitlab.ds.unipi.gr/interopehrate/reference-hcp-app/terminal-d2d-hr-exchange/tree/security
3. Open the cloned project using the Maven opener menu of the desired IDE.
4. Build the project by generating the jar file and ensure that is located in the Maven repository of libraries of the computer that the HCP App project will be located.
5. Include the necessary dependencies for the D2D library jar file inside the HCP App project.

3.2.2. User guide

This user guide refers to the version 1.0 (V1) of the IDM functionalities of the HCP side as an extension of the D2D library. The main flow of the functionality of this library is the following:

- The `fetchCertificate()` function will be called to acquire from the CA the necessary Certificates and store them to the `Keystore`.

```
public void fetchCertificate() throws IOException,
NoSuchAlgorithmException, KeyStoreException,
CertificateException, UnrecoverableEntryException;
```

- In addition to the Bluetooth MAC address of the HCP app running machine, the signed MAC address will also be created. The function returns the concatenation of the following three Strings: Bluetooth MAC address + # + Signature

```
public String signPayload(String payload, PrivateKey privateKey)
throws IOException, SignatureException, InvalidKeyException,
NoSuchAlgorithmException, InvalidKeySpecException;
```

- The `sendHCPCertificate`() function is called immediately after the connection is established and sends to the S-EHR App the following two concatenated Strings: pubkey + Base64 representation of the Certificate

```
public void sendHCPCertificate()

throws IOException, CertificateException,
NoSuchAlgorithmException, KeyStoreException,
UnrecoverableEntryException, InvalidKeySpecException;
```

- When the Certificate is received from the S-EHR app, we call the `onCertRecivedStore()` function to store the Certificate to the KeyStore. This function returns true upon successful Certificate storage and false in case of failure.

```
public boolean onCertRecivedStore(Certificate cert)
```

REFERENCES

[1] **[D3.3]** Specification of remote and D2D IDM mechanisms for HRs Interoperability - V1, 2019. https://www.interopehrate.eu/resources/#dels

[2] **[D3.9]** InteropEHRate Consortium, Design of libraries for HR security and privacy services - V1, 2019. https://www.interopehrate.eu/resources/#dels

[3] **[D4.12]** InteropEHRate Consortium, Libraries for remote and D2D HR exchange - V1, 2019. https://www.interopehrate.eu/resources/#dels