Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 58(72), Fascicola 1, 2013

Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario

D. Popa[1] K. Boudaoud[2] M. Cremene[1] M. Borda[1]

Abstract – Mobile Cloud Computing triggered the implementation of several applications that easily provide the users with more complex features and characteristics. Secure Mobile-Cloud Framework is designed in order to secure users private data transmitted between the same mobile cloud application components. This work will present a solution to integrate the Secure Mobile-Cloud Framework into a healthcare application scenario.
Keywords: Mobile Cloud Computing, Security, Mobile Cloud Applications

I. INTRODUCTION

Mobile Cloud Computing [1] is a new concept, which offers Cloud Computing resources and services to mobile devices.
Cloud Computing [2] is a new technology that provides, to the Internet users, data, resources, platforms, and applications as services.
In the last years mobile phones have greatly developed, and because of the small size and also because they can be moved easily, they become indispensable to users.
Using Mobile Cloud Computing advantages, new application models were developed for mobile devices. These applications models try to use both the mobile device resources and the Cloud services to provide a more reach and a more varied functionality in order to increase the mobile device popularity and use.
From a security point of view, Mobile Cloud Computing, increases the security risks and privacy invasion due to the fact that it combines mobile devices with Cloud services and also because there is not a well-defined application model [3].
The security issues are treated independently and the existing security solutions are supplied separately by various providers (Cloud providers or mobile platforms). Thereby, in the case of mobile cloud applications, it is needed to combine different solutions in order to secure them. Also, very few solution proposed to secure the data or the applications take into account the mobile device energy constraints, users constraints or data sensitivity constraints.
The Secure Mobile-Cloud Framework proposed in [4] focuses on component based mobile cloud applications. Its goal is to secure the communication between the application components. The feature of this solution is the fact that it tacks into consideration the following constraints: mobile device energy, data sensitivity and users' options. The security framework is composed of components deployed on the mobile device and components deployed in Cloud.
In this paper we discuss the integration of the security framework into a mobile cloud application scenario.
The paper is organized as follow. Section II briefly describes the security framework. In Section III, it is presented the integration solution. This section also describes the design of a mobile cloud application scenario. Finally, in section IV, several conclusions are presented.

II. SECURE MOBILE-CLOUD FRAMEWORK SHORT OVERVIEW

The Secure Mobile Cloud (SMC) framework is composed of two types of components, as presented in [5]: 1) security components and 2) management components.
The security components have been designed in [6] for the LECCSAM architecture. Their role is to implement the eponym security properties (e.g. integrity, authenticity, confidentiality, non-repudiation). These security components are deployed in both mobile device and Cloud. The management components have been designed to identify and apply the appropriate security properties and therefore security components to users' data. Some of these management components are deployed on the mobile device and some of them are deployed in the Cloud.

We want to allow the users to express their choices regarding the security level they want to apply to their data. In order to provide a solution that allows achieving this characteristic there was designed an analysis system. This system is part of the security

[1] Technical University of Cluj-Napoca, Communications Department,
Str. Dorobantilor. 71-73 CP. 400609 Cluj-Napoca, Romania, Daniela.Popa @com.utcluj.ro
[2]University of Nice Sophia Antipolis,
930 Route des Colles - BP 145- 06903 Sophia Antipolis

framework. And it has to provide the security combination needed to be applied to data (security combination = security properties + security algorithms); and also the location where this combination can be performed (e.g. on the mobile or in Cloud).
A briefly description of the management components is given in Table 1.

Table 1 Description of Managers

Manager	Description
Mobile Manager	It collects data and events that occurs on the mobile side and sends them to the appropriate manager to be analyzed.
Mobile Security Manager Cloud Security Manager	Both provide the composition of the security properties. The Mobile Security Manager ensures security composition on the mobile device side and the Cloud Security Manager ensures the composition on the Cloud side.
Optimization Manager	It sends the information collected from sensors (e.g. network sensor, energy sensor) to the mobile manager.
Policy Manager	It determinates which security components are required for a specific security level.

III. THE INTEGRATION

The term integration denotes the combination of the mobile cloud application scenario with the security framework. Furthermore, it refers to the way in which the application scenario will operate using the security framework.
For the integration solution it is assumed that there is access to the application scenario source code. The proposed solution for the integration is a static solution. A static solution assumes the integration at the application source code level.
It is also assumed that the keys (public and secret) are securely distributed.
This section is split in two parts. The first part presents the application scenario design and the second part describes the integration solution.

A. *The mobile cloud application scenario*

The proposed application scenario is a healthcare application. This application aims to monitor a person type according to his/hers bodies characteristics and to propose a regime. The application initial characteristics and functionalities are presented in the following.
The application captures the users' characteristics: body size (e.g. height, weight). According to the information received, a type of user is established. Depending of the user type, a certain regime is provided.
The application functionalities from the user point of view are (see also Fig.1):
1. To create an account, when a user is using the application for the first time. In order to do this the user must provide several private data. This private data are: the name, password, e-mail.
2. If the user already has an account he/she needs to login.
3. Insert the user data, which may be of two types:
   Personal data: day of birth and gender
   Body measurements: height and weight
4. To obtain the user type. The user type is established by users' personal data and users' body size data. There are defined four types of users: weight-gain users, weight-loss users and normal users. The weight-gain users are too skinny and that they have to take in weight. The weight-loss users are too corporal and they have too loose weight. The normal users do not need to change their body size.
5. To obtain the regime. The regime denotes the types of food the user can or cannot consume.

This application scenario is intended to be a Mobile Cloud Computing application, based on components running on the mobile device side or in Cloud.
The application was split in three major parts. These parts are:
- *The user interface*. On the mobile device. It is designed in order to collect the users' data.
- *The database*. On the Cloud side. It is designed in order to store the data.
- *The services*. On the Cloud side. There are designed in order to execute the following application functionalities: compute the user type, compute the BMI (body max index), and compute the regime. For each functionality a service was designed.

As it can be seen in Fig.2 the user interface was

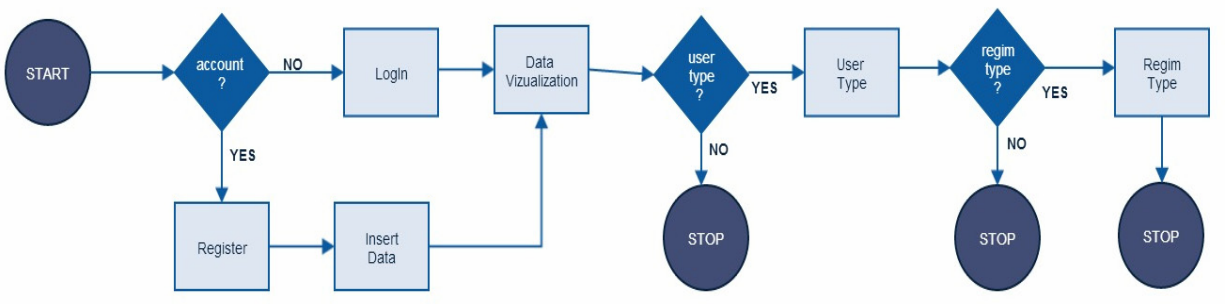

Fig.1 Application Scenario functionality

designed to be a component; the database was also designed as another component; and each service represent a different component.

B. *The integration solution – theoretical approach*

As previously said the integration assumes the combination of the application scenario with the security framework. In this way it is obtained a new application, the application scenario secured; an application that has the functionalities of the application scenario, but that it also has the data security ensured.
The problems that need to be resolved here are the following: 1) where and how the application scenario will call the security framework to secure the data; 2) which is the flow of the application.

As solution to the first issue, for each component, on the mobile device (user interface) or in Cloud (services), it was designed a part, called ApplySecurity, in order to communicate with Mobile Manager, in the case of the user interface, or with Cloud Security Manager in the case of a service in Cloud.

A solution to the second issue is described below:
From the description of the scenario application in the previous section it can be seen that the application includes three types of communications: 1) the communication between the mobile device and the database; 2) the communication between the mobile device and the services; and 3) the communication between two services.

The communication between the mobile device and the database assumes the following actions: 1) saving data provided by the application user into the database; and 2) requesting data from the database in order to be displayed on the application interface.
The communication between the mobile device and the services assumes calling the services and providing the proper users' private data in order to obtain a certain result; the result may be or not a users' private data.
The communications between two services refers to one service calling the other services and providing the proper data (private or not) in order to obtain certain data as result (private or not).

In this work it will be presented the secured flow for the communication between the mobile device and the database.
The secured application scenario flow in the case of communication between the mobile device and the database requires several steps (see Fig. 3). These steps are:
S-1. Users' private data, retrieved by the user interface, which need to be saved into the Cloud database are intercepted by the ApplySecurity part and sent to the Mobile Manager along with their sensibility level;
S-2. The Mobile Manager, uses its analysis functionality to verify where is more suitable to apply the security, on the mobile device or in the Cloud;
S-3. If the mobile device is chosen by the Mobile Manager, the data along with the security level (the combination of security properties along with the security algorithms) are sent to the Mobile Security

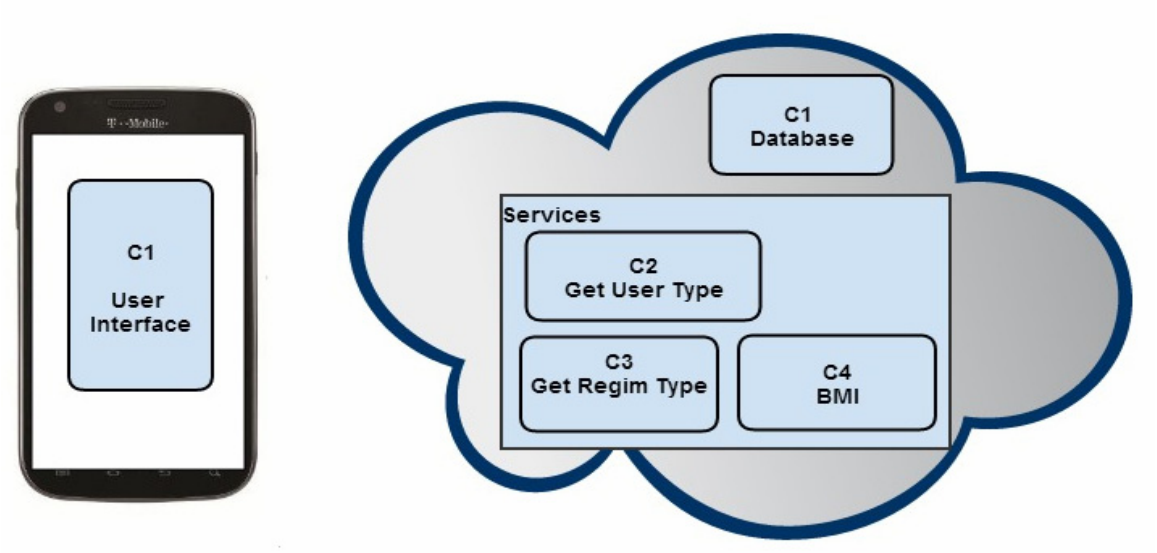

Fig.2 Application Scenario Components

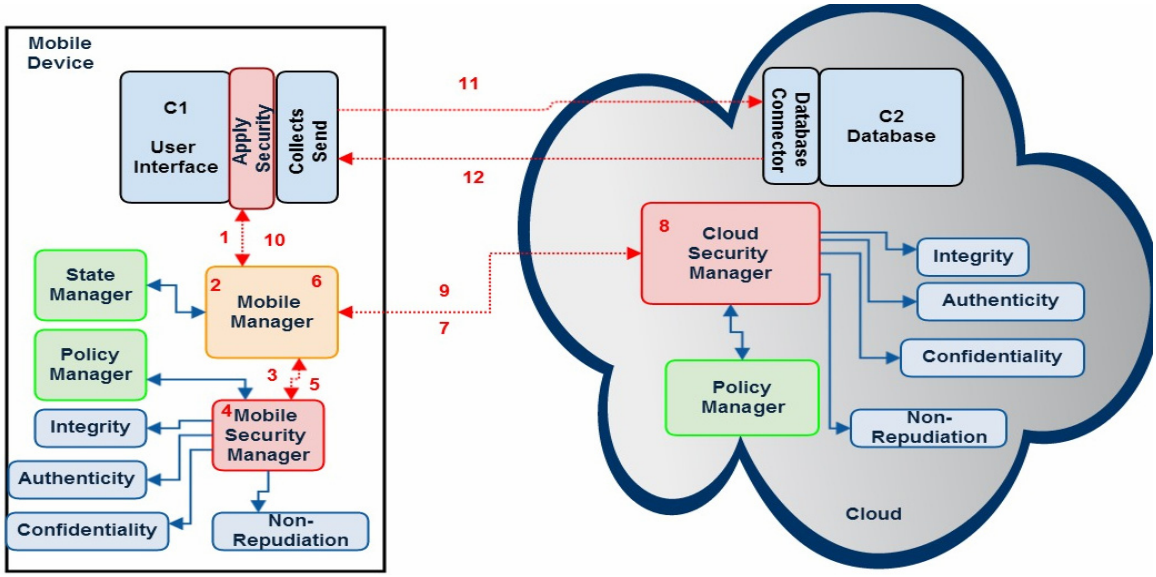

Fig.3 Secured Application Scenario Flow

Manager;
S-4. After discovering which security properties and security algorithms correspond to the received security level, the Mobile Security Manager orchestrates the application of the appropriate security properties (components) to the received data;
S-5. When the security operation is finished, the secured data are sent back to Mobile Manager;
S-10. The secured data along with the security level are sent to the ApplySecurity part;
S-11. The secured data are sent to the Cloud database;
S-12. A response message of success or of error is received;

S-6. If the Cloud is chosen by the Mobile Manager, data along with the security level are ciphered using a secret key;
S-7. Encrypted data are sent to the Cloud Security Manager;
S-8. After decrypting the received data, Cloud Security Manager orchestrates the application of the appropriate security properties (components) to the received data;
S-9. When the security operation is finished, the secured data are sent back to Mobile Manager;
Then, the steps from S-10 to S-12 are followed.

C. *The integration solution – technical approach*

This section includes the application scenario implementation and it shows from a technical point of view where in the application scenario it must to intervene in order to add the security framework.
At this time from the application scenario it has been implemented only the communication between the mobile device and a database stored in Cloud.
The main languages that were used to implement the application scenario are Java, MySQL, PHP and JavaScript Object Notation (JSON).

The interface on the mobile device was implemented using Java based on Android, PHP was used in order to access and query the database and for the database implementation was used MySQL. Then, in order to make a connection between the mobile device and the server the JSON data-interchange format was used.
For the communication between the mobile device and a database in Cloud the application design consists of: the user interface, the collect/send part, the connector part and the database. Thus, the user interface consists of three screens: LogIn screen, Register screen and Details screen. The collect/send part consists of two classes implemented in Java: UserFunctons and JSONParser. The connector part consists of two files: index and DBFunctions implemented in PHP. The database is implemented in MySQL.

The UserFunctions class was implemented in order to send the user data, retrieved from the user interface, to the Cloud database. It implements the following methods:
- loginUser(): it implements the login requests
- registerUser(): it register the user login details (name, e-mail and password) into the external database
- registerUserInfo(): it register the user other info like: gender, height, weight, day of birth into the external database

The JSONParser class implements the following method:
- getJSONFromUrl(String url, List<NameValuePair> params): it passes and receive the data from and to the server side implemented in PHP.

The index file handles all the request coming from the mobile device. The DBFunctions include the implementation for DBFunctions class methods. The class implements two methods: storeUser and storeNewUserInfo. Their role is to implement the operations of insert into the database.
The database contains two tables:

- users: to store the register data: name, e-mail and password
- info: to store the users information like gender, height, weight and day of birth

The following section will present: 1) how there were defined and integrated, into the application code, the constraints that the application designer should specify; 2) in what consist the ApplySecurity part mentioned in Theoretical approach section; and 3) the modifications made to the user interface.
The constraints that the application architect needs to specify are the application type and the data sensitivity level.
For the application type constraint, the solution applied was to use in the application manifest file the attribute 'description' (see Fig. 4). This description attribute will take the application type value; for this particular application scenario the value is health. To retrieve the attribute description value, it was implemented into the main user interface page (UIMainActivity.java) a method called getApplicationType (see Fig. 4). The getApplicationType method uses the style resource identifier getApplicationInfo.descriptionRes, provided by android.content.Context in order to identify in the application resources the attribute description value.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:description="@string/app_type"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.app.sampleUI.StartApiSampleActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

app_name = getApplicationName(getApplicationContext());
app_type = getApplicationType(getApplicationContext());

public static String getApplicationType(Context context) {
    int stringId = context.getApplicationInfo().descriptionRes;
    return context.getString(stringId);
}
```

Fig.4 The application type constraint

For the sensitivity level constraint, the solution applied was to define several private static variables into the class (UserFunctions) that is in charge of the data transmission. These variable are: add_security, highS, mediumS and lowS as it can be seen in Fig.5.

```
/*for security*/
private static String add_security = "add_security";
private static String highS = "high";
private static String mediumS = "medium";
private static String lowS = "low";
/*for security*/
```

Fig.5 Sensitivity level constraint - solution

In the theoretical approach section was mentioned the ApplySecurity part whit the functionality of collecting the user private data and sending them to the Mobile Manager.
ApplySecurity part and its functionality were implemented into each UserFunction method that sends data to the database as follow:
While the list whit pairs of type <parameter,value> is build, it is inserted before each pair an additional pair of type <S_level, value> as it can be seen in the Fig. 6. After the list is built, the apply_security_encrypt() method in MobileManagerPartB class is called. This method will return a list of type <parameter,value>, where the values are secured.
Then the data may be sent to the database.

```
params.add(new BasicNameValuePair("security", add_security));
params.add(new BasicNameValuePair("tag", register_tag));
params.add(new BasicNameValuePair("num_reg", num_regist1));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("name", name));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("email", email));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("password", password));

params_json = SMC_manager.apply_security_encrypt(context,db_app_name,params);

JSONObject json = jsonParser.getJSONFromUrl(registerURL, params_json);
```

Fig.6 ApplySecurity Part

The apply_security_encrypt() method has behind it the functionality provided by the security framework SMC such as:

- It is determined the security combination needed depending on: users' options, data sensitivity level and mobile device energy level;
- It is applied the security combination. The security combination is the security components combination designed as part of SMC Framework.

The security algorithms used in the SMC Framework are presented in Table 2.

Table 2 Algorithms

Algorithm Type	Algorithm Name
Symmetric	DES, AES-128, AES-256
Asymmetric	RSA
Hash	SHA-1, SHA-256

For the user interface modification (see Fig. 7); it was introduce a new main user interface from which the user may choose to set the security settings or to start the application. In the case the user won't set the security settings, it will be applied to all data without taking into consideration the data sensitivity level, the default security level for a standard user type.

In Fig.8 can be seen the users' private data saved into the database after performing the Register action. The data are secured.

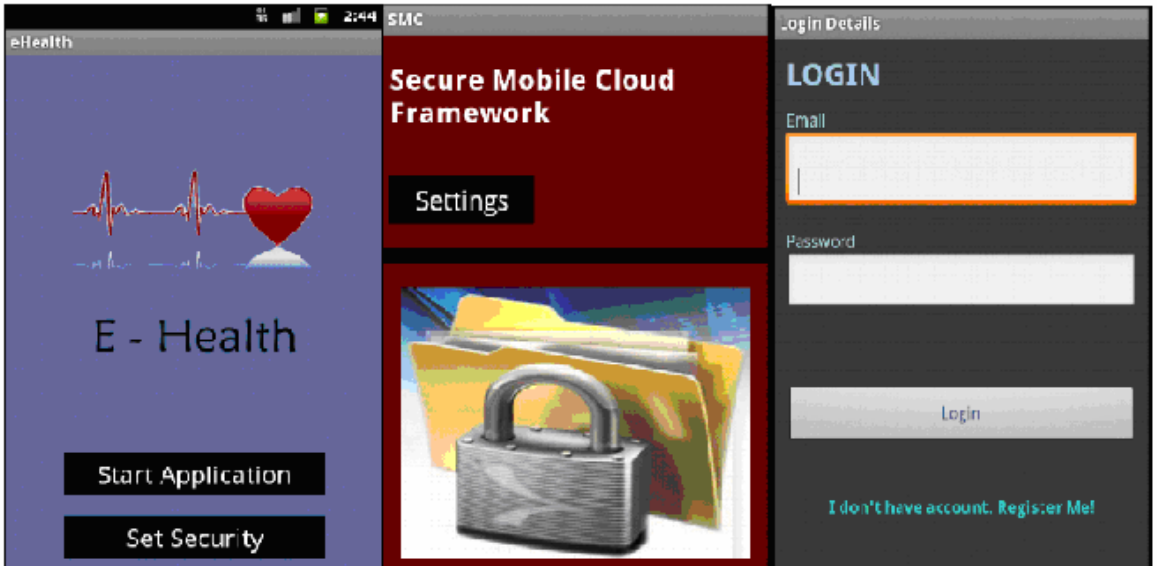

Fig.7 User Interface

name	email	password
xNFt1zLE5EFtyoj4PqWDJVsFurXIS/	DB6MsfPaARvGuXJ1urqfK1RWiOvLtx	14x/dElahyZh2WTuNHUVYSCjxDYK4y

Fig.8 Users' private data save into the database

V. CONCLUSIONS

A healthcare mobile cloud application scenario was presented in this paper. The application aims to monitor a person type according to her/his bodies characteristics, and then to propose a regime. The application was split into components. The user interface was designed as a mobile device component, while the database was designed as a Cloud component. Three services running as components in the Cloud were also designed.

In this work was also presented a solution to integrate the Secure Mobile-Cloud Framework into the healthcare application scenario. The intended result was to obtain only one application. The new obtained application has the healthcare application functionalities, but it also has the security of data transmitted between components. The integration is done at the source code level.

For the integration solution, two approached are discussed, the theoretical approach and the technical approach. The theoretical approach shows where in the application scenario flow it is needed to be done the call to the Secure Mobile-Cloud Framework components. The technical approach shows the modifications brought to the application scenario source code in order to integrate the security framework.

The drawback of the integration solution proposed in this work is the fact that it needs the application scenario source code. As future work, this solution can be improved and changed into a dynamic one. A dynamic solution that does not needs the source code and which can be applied for any component based mobile cloud application.

ACKNOWLEDGMENT

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

REFERENCES

[1] B.G. Chun and P. Maniatis, "Augmented Smartphone Applications Through Clone Cloud Execution," in Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII), USENIX, 2009.

[2] Jeremy Geelan. "Twenty one experts define cloud computing", Electronic Magazine, available online: http://virtualization.sys-con.com/node/612375, August 2008.

[3] C. Nachenberg, "A Window Into Mobile Device Security – Examining the security approaches employed in Apple's iOS and Google's Android", Symantec Security Response.

[4] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications", in Proceedings ROEduNet 11 th International Conference, Sinaia, 2013.

[5] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", The 6th International Conference on Security for Information Technology and Communications, June 25, 2013.

[6] M. Kamel, K. Boudaoud, S. Resondry and M. Riveill, Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments, in Proceedings of the 12th IFIP/IEEE, Dublin, Ireland, May, 2011.