Acquisition of Confidential Patient Data Over Shared Mobile Device

Petr Vcelak[1,2], Martin Kryl[2], Ladislav Racak[2] and Jana Kleckova[1,2]

[1] *NTIS – New Technologies for the Information Society, University of West Bohemia, Univerzitni 8, Plzen, Czech Republic*

[2] *Department of Computer Science and Engineering, University of West Bohemia, Univerzitni 8, Plzen, Czech Republic*

Keywords: Client-server, Data Acquisition, Mobile Application, Patient Data, Security, Shared Mobile Device, Smartphones, Tablets.

Abstract: Mobile devices have already been designed for many applications. Smartphones and tablet computers are modern, widespread and affordable solutions used for various purposes. Nowadays mobile devices are widely used in telemedicine. It is usually assumed, that the device is owned and used by a single person. We focus on security concerns and constraints from a different point of view – when the device is shared. In this paper, we are proposing a novel approach to prevent leakage of patient's confidential data when the device is used by multiple patients at the hospital's clinic or department. We present a prototype application and discuss its use case and designed workflow.

1 INTRODUCTION

Our goal is to enable gathering of patient's data via shared mobile device. Hospital would lend the configured device to patients and they would be able to fill out pre-defined medical questionnaires and personal information forms while waiting for their medical examination in situation, when medical personnel is busy examining another patient. Some types of data can be conveniently collected at this time. The doctor would then pass and verify data, possibly complementing deficiencies. It is not just about gathering general information like address, contact or health insurance situation. Forms could be used to gather more information about patient's perceived discomfort during the last period. Patient can also provide data for surveys, data for conducting studies or use mobile devices to measure additional data such as weight, blood pressure or pulse through wearable electronics. At this time, we do not expect the use of additional equipment like wearable electronics and sensors.

2 STATE OF THE ART

The main target in this paper are mobile phones such as smartphones and electronic tablets. Nowadays mobile phones are used in all healthcare areas including diagnostics, telemedicine, research, reference libraries and interventions. (Bastawrous and Armstrong, 2013) Currently, these devices are easily available, inexpensive, small, have enough computing power and provide sufficient space for the development of various types of new applications. An advantage is a user-friendly interface and availability of installed applications.

Different mobile applications are widely used in medicine with the aim to provide personalised approach or just for gathering health data. As Hayes et al. (2014) said, there were areas where patient-tailored risk prediction and treatment had been applied routinely in the clinic over mobile applications. Nevertheless, authors said, more work would be required to translate scientific advances into individualised treatment in other fields. (Hayes et al., 2014) There were publications regarding that, eg. a smartphone-centric platform for remote health monitoring of health failure (Bisio et al., 2015) or cloud-based smart health monitoring system for automatic cardiovascular and fall risk assessment in hypertensive patients (Melillo et al., 2015).

Android is the best selling operation system on tablets since 2013, and on smartphones it is dominant by any metric. (Manjoo, 2015) We can cite many articles and examples describing data acquisition via mobile devices on the Android platform, eg. a portable physical health monitoring system were proposed in (Tang et al., 2015) and continuous wireless monitoring of endogenous and exogenous bio-molecules on an android interface in (Stradolini et al., 2015).

Vcelak P., Kryl M., Racak L. and Kleckova J.
Acquisition of Confidential Patient Data Over Shared Mobile Device.
DOI: 10.5220/0006155303340339
In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)*, pages 334-339
ISBN: 978-989-758-213-4

On the other side, security and privacy issues are serious topics. Authors Baig et al. discussed mobile healthcare applications and its critical issues and challenges. As they said in (Baig et al., 2015), mobile phones were becoming important in monitoring and even in delivering of healthcare interventions. Results of testing mobile health applications on Android platform was discussed in (Knorr and Aspinall, 2015) where a number of serious vulnerabilities were discovered in the most popular applications. Dehling et al. (2015) did an overview of security and privacy infringements in mobile health applications on Android and iOS. They discovered that the majority of apps (95.63%, 17,193/17,979; of apps) had posed at least some potential damage through information security and privacy infringements. There were 11.67% (2,098/17,979) of apps that scored the highest assessments of potential damages. (Dehling et al., 2015) These results lead to belief that private or confidential information stored in a mobile device/application are at a risk.

3 SHARED DEVICE APPROACH

3.1 General Information

An obtaining data at own mobile device is an usual approach. Unfortunately, this may not be a safe way. Unlike the usual situation where everyone has their own mobile phone, we start from the opposite assumption. We consider the use of device that is not owned by the patient. It is only borrowed at a given moment. We refer to this as a **shared device** (SD). The patient has to fill new information or update existing data (delivered to/from the information system of the hospital or medical doctor) by the shared device.

An advantage of the SD approach is the ability to fully control its system and customise it. We can set up the environment and install all necessary tools, including our own custom applications. Through our custom application we can provide personalising. It can be determined in advance what information or questionnaire needs to be filled by particular patient. Different type and extent of data might be desired for different departments, patients, diagnosis or type of visit.

After the registration at the desk/office/nurse, patient can get pre-configured shared device, and can immediately begin checking and filling the form. Optionally, patient may be allowed to switch to other applications, eg. read news or play some simple games, while still waiting for examination.

Disadvantages of shared devices are especially the need to solve the issue of patient's privacy. Is it really a disadvantage though? Sensitive or protected data must not be available to the next patient/personnel using the device. We have to prevent data leaks when someone steals the shared device. Of course, data transfer have to be secured. What about other health mobile applications? Any kind of health application has to secure its data as a prevention to data leak, otherwise it leads to security and privacy infringements. Well, the data privacy have to be solved with any kind of application that works with patient health information. The difference for shared device approach is, that we are absolutely sure the application will be used by multiple users, therefore it has to be secured better.

3.2 Architecture

We have chosen client-server architecture where client will be responsible for a user interaction. The business logic will be on the server side only, eg. form definition and description, form source/input data and produced output. Client-server communication is based on REST API with public and private key-pairs encrypted messages. The HTTPS protocol is recommended.

3.2.1 Client

Client, as a mobile application, will provide direct interaction with users. User can be a staff member or a patient itself. We prefer to make client application as simple as possible. In the figure 1 you can see use case diagram. We expect no local configuration stored in a shared device with the exception of the URL server address. Security details are described in a separate section 3.4. Basically, the client application is prepared when URL address of the server is set. The shared device can operate in two modes:

user have to authenticate by username and password knowledge,

delegated no user authentication; server sends available forms and identifies retrieved data by device ID.

In user mode, there is no list of available users on SD for a user authentication. Instead an encrypted request (with username and password) is sent to server for authentication. After a user is authenticated, forms (only available to the user) are downloaded , and the user can choose either to be filled. At the same time the user can also choose to continue filling up an unfinished form, that has not been uploaded yet. Client

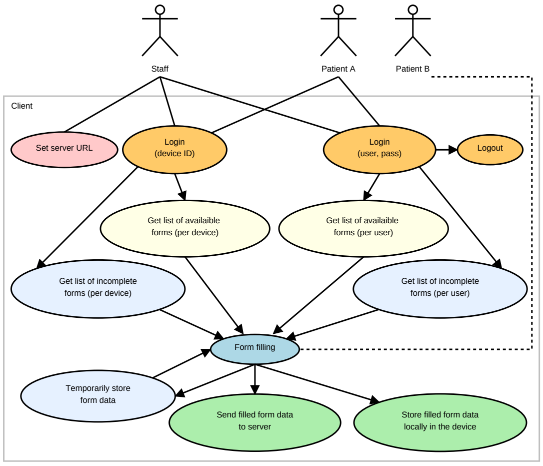

Staff
Patient A
Patient B
Client
Set server URL
Login (device ID)
Login (user, pass)
Logout
Get list of availaible forms (per device)
Get list of availaible forms (per user)
Get list of incomplete forms (per device)
Get list of incomplete forms (per user)
Form filling
Temporarily store form data
Send filled form data to server
Store filled form data locally in the device

Figure 1: Use case of client.

requests list of available forms for the device directly when delegated mode is used.

List of available forms is user-dependent and device-dependent to allow for possible customisation. This is a quite simple solution for delivering different forms to the same patient on different clinics.

In the figure 1 you can see use case of delegated mode by *Patient B* actor. Shared device is prepared by Staff (actor) by using *Login (device ID)* method and the patient just fills out the form.

The *Temporarily store form data* use case can resolve any distraction (eg. the need to go to the toilet). Later, a patient can use *Get list of incomplete forms* use case (stored per device or per user).

Finally, user can finish completing form by two methods: (1) send data to the server or (2) store results data locally in the device. It is preferred to directly send data to the server. The second method is a backup solution, eg. when Wi-Fi signal is lost. Locally stored data is encrypted.

3.2.2 Server

REST API provided by server is the most important from the perspective of the client. There in the figure 2 you can see all of our actors. A client application represents both the top use cases, *User (logged)* and *User (not logged)* in the delegated mode. Both actors can get/download server's public key S_{pub}, list of available forms and upload/store filled form data. The form content can be uploaded online over Wi-Fi network or offline by connecting device by a cable.

Server must be properly secured on an operation system level. All retrieved data from the client applications will be stored there in an unencrypted form. For simplicity's sake, we do not describe details of server configuration, which may be implementation-dependent and include all activities related to the administration of users, forms and a description of their assignment to users or devices. In this context, *Administrator* and *Other System* actors use cases are out of scope of this paper.

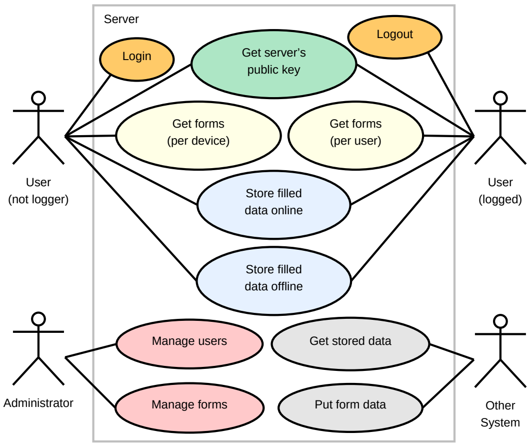


Figure 2: Use case of server.

3.3 Form Description

There are several ways to describe the form elements, their labels, groups, description and all the essentials including, eg. validation or enumeration values. There exist different forms description languages like XForms 1.1 (2009) and different libraries that helps building forms. In addition, there are number of differences across mobile platforms. We have made our own way of simplified definition and description of forms. The same form will need to be viewed and filled-in on different platforms. Our simplified definition and description of forms contains:

- form description – ID, name and description
- section – group of form elements has title,
- element – ID, label, data type, validation, default value, enumeration values,

All text labels may occur multiple times with different language attribute. Supported element data types are label, text, multi-line text, email, password, integer, number, currency, phone, boolean check or switch, date, time, selector box, selector list and URL. The form definition (description and content) have to be rendered by client application per mobile platform.

3.4 User Authentication Model Definition

The most important part of the shared device approach is communication protocol between client and

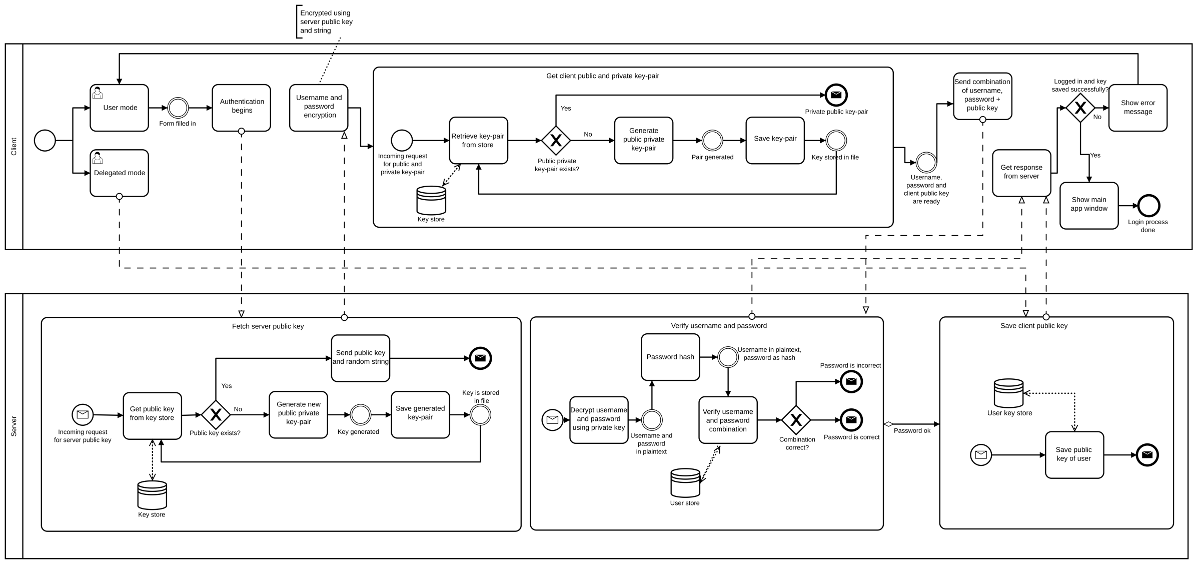

Encrypted using
server public key
and string
Client
User mode
Form filled in
Authentication begins
Username and password encryption
Delegated mode
Get client public and private key-pair
Incoming request for public and private key-pair
Retrieve key-pair from store
Key store
Yes
No
Public private key-pair exists?
Generate public private key-pair
Pair generated
Save key-pair
Key stored in file
Private public key-pair
Username, password and client public key are ready
Send combination of username, password + public key
Get response from server
Logged in and key saved successfully?
No
Yes
Show error message
Show main app window
Login process done
Server
Fetch server public key
Incoming request for server public key
Get public key from key store
Public key exists?
Yes
No
Send public key and random string
Generate new public private key-pair
Key generated
Save generated key-pair
Key is stored in file
Key store
Verify username and password
Decrypt username and password using private key
Username and password in plaintext
Password hash
Username in plaintext, password as hash
Verify username and password combination
User store
Combination correct?
Password is incorrect
Password is correct
Password ok
Save client public key
User key store
Save public key of user

Figure 3: Diagram of the login process – BPMN v2.0 notation.

server. We need to ensure that data in the client application is protected against unauthorised access (read) or modification at the device or during transmission to the server. For this reason, we describe how user login process works on mobile devices. You can see the login model in the figure 3. Based on the model, encryption is enabled for stored data on client and communications between client and server.

As we mentioned in 3.2.1 and 3.2.2 sections, client application can login in two ways.

User does not need any username or password in the *delegated mode*, unique device ID is used instead. Private and public keys are required on the device. New key-pair (C_{priv} and C_{pub}) is generated and stored in an internal memory of mobile device when no key-pair exist. Both, device ID D and public key C_{pub} are sent to the server. The client may see a list of currently-fill forms if the server successfully processes the request, so device identification and its public key are stored on a server. If communication with server fails, the client is informed of the error message and remains on the login screen.

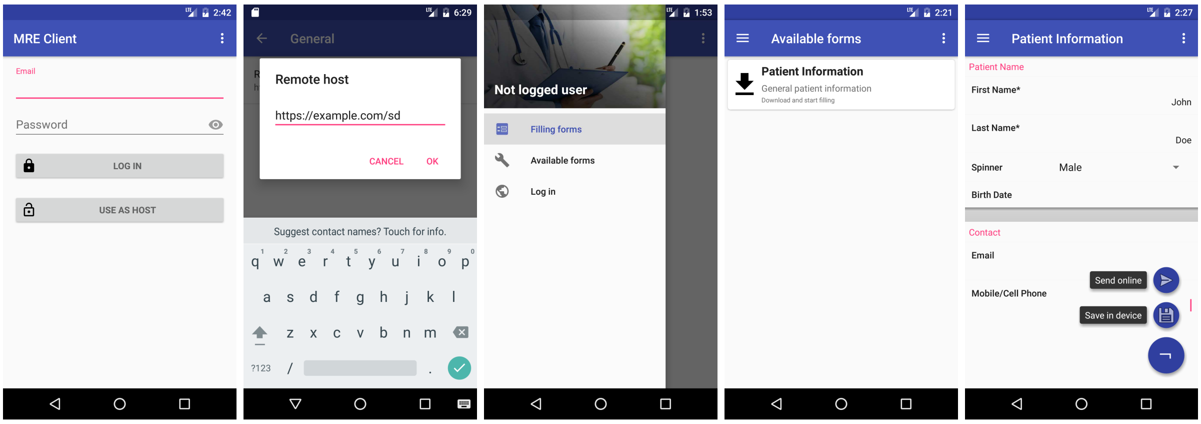
A user login using username U and password P method need to have a public key of the server S_{pub}. Client application asks for server's public key S_{pub}. Server immediately sends an existing key or generates a new key pair by a configured algorithm. A cryptographic salt (CS) is sent together with the public key S_{pub}. Client stores both in an internal memory. Authentication then follows. Client application encrypts username U and password P by server's public key S_{pub} and salt (CS) added. Server receives data and decrypts them by its own S_{priv} key. Server checks that the received salt matches the salt sent to the device. Request is denied if the salt differs or the user does not exist on the server. A password hash stored on the server is compared with the one received from the client. The client's public key C_{pub} is stored in case of the same password hashes. Client is notified that the user has logged in successfully and shows list of possible forms. Otherwise, client is informed of the error and remains on the login screen.

The salt (CS) is device-dependent. User's key-pair (C_{priv}, C_{pub}) depends on a username U and device ID D. Each pair of user (as username U) and device ID D has its own key-pair because of the same key reuse.

4 IMPLEMENTATION AND RESULTS

As the proof of concept we have created software corresponding to designed client-server architecture (3.2) and supporting the user authentication model (3.4). Server is a service-oriented Java web application with REST API and running over HTTPS protocol. The server needs Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files installed. We chose the Android platform, the world's most popular mobile operating system, for the client application. The implementation is based on Android version 7.0 with code name Nougat (API level 24). The five screenshots are shown on the figure 4.

The client application running on a shared device does not have any complicated setting, and provides only a few basic functions. The only configuration attribute is the server URL address. Mobile applica-



Remote host
https://example.com/sd

Patient Information
General patient information
Download and start filling

Patient Name
First Name* John
Last Name* Doe
Spinner Male
Birth Date

Figure 4: Mobile application (client) screenshots: (1) login screen, (2) configuration – set server URL address, (3) menu of actions in the delegated mode, user can fill-in per device forms or not finished ones, (4) list of available forms and (5) simple form/questionnaire generated and menu how to submit a result is shown.

tion has to download corresponding definitions of the forms that the user can fill in on the shared device. User cannot directly access data of any hospital information systems. Shared device is managed indirectly by the staff through the server. Staff member can assign a form type to the specific device or patient on the server. Client application downloads these form definitions and present an automatically generated forms to its user.

Retrieved, transmitted and permanently stored data on the shared device are encrypted. Only the patient himself can see his own data in a readable way. Form content is encrypted immediately, when form is validated and submitted. Online transmission is the preferred way of data transmission, but encrypted data can be stored locally when no network access is available. Data encryption is based on an asymmetric method using private and public key-pairs that are generated on the server or on the shared device.

5 DISCUSSION

We discuss the results and especially the security of the proposed shared device approach in this chapter. The most common and possible attacks include eavesdropping or modification of data and identity theft.

Client-server communication is designed to encrypt the whole communication, including form definitions, because it can contain personal information. The only unencrypted content is server's public key, client's public key and device ID. An attacker can get access to communication content on an unsecured network (eg. public Wi-Fi, HTTP protocol), but it is not possible to abuse this, because of the way asymmetric cryptography works (private key is needed for decryption). Eavesdropping is therefore not possible.

An another attack is by modifying request to obtain data access. It can happen on an unsecured network or when a shared device is infected by malware. Attacker can spoof the device ID via malware and request forms available for spoofed ID. Server sends forms belonging to someone else, however, they are not accessible, because the private key on attacker's device cannot de-crypt them. Attacker would need to possess private key of the device that is being imitated by spoofed device ID.

An entire request could be alternated by attacker, not only the device ID. Server process every request if its format is valid. Attack would be discovered, because server attempts to verify the electronic signature using the sender's public key. This verification would fail. Either electronic signature and request content or public and private key-pair will differ.

We also consider the case of identity theft. Attacker can monitor requests and reuse the message to repeat sending of that request with own fake public key. This situation is prevented by changing cryptographic salt, which is always used to encrypt user name and password. Accordingly, the resulting cipher names and passwords vary because of the different cryptographic salt.

Our, originally naive, solution of form definition/description proved to be very flexible. The form is dynamically generated on the mobile device (client), according to these definition. It is possible to dynamically create personalised form definition on the server which contain patient-oriented data (eg. different options per age groups or genders).

Mobile devices also have drawbacks. The client application needs to be online when loading the form definition. There are also issues with battery life, network availability, breakage or damage of borrowed device by patients. Network availability is important when submitting form data to the server, but we can use offline data storage to temporarily alleviate connection problems.

At the testing phase, we identified an issue (occurred only once) with locally stored encrypted form content. There is a possibility of data loss, when user logs out of application and logs in again, because key-pair has to change and server does not store history of previously used keys. Our approach, originally devised to prevent attacks, may therefore lead to data loss on this occasion.

The mobile application has fully satisfactory and fluent response when connected via Wi-Fi. Only when using large forms (e.g. 1 000 items in select box) the response time worsened – increased time needed to download form definition and to render the form.

6 CONCLUSIONS

In this article we proposed an approach for secure health data acquisition using shared mobile devices. The data are confidential in general. The primary goal was gathering personal data and updating health status using form-oriented application. The security problems were discussed. We evaluated risk of data leak and designed data workflow for mobile devices that are shared across patients.

We designed a prototype and evaluated it as the real application on Android device. We identified that this workflow is properly secured. The discovered disadvantage is a possible data loss in special case when data were encrypted and key-pair changed before data was delivered to the server.

Obtaining data via electronic forms is easily customisable and extensible. There is a potential disadvantage of impersonal approach. Though, patients waiting for medical examination are usually feeling bored, and this interactive form might be therefore appreciated by them.

In the future we plan to expand the types of data that can be sent via secure forms and the presented approach. The logical extension is to support wearable electronics, sensors and other accessories connectable with a mobile device that will also acquire more data types in this way.

ACKNOWLEDGEMENTS

This work was supported by the project PUNTIS (No. LO1506) of the Ministry of Education, Youth and Sports of the Czech Republic and by University of West Bohemia, project Data and Software Engineering for Advanced Applications (No. SGS-2016-018).

REFERENCES

Baig, M. M., GholamHosseini, H., and Connolly, M. J. (2015). Mobile healthcare applications: system design review, critical issues and challenges. *Australasian Physical & Engineering Sciences in Medicine*, 38(1):23–38.

Bastawrous, A. and Armstrong, M. J. (2013). Mobile health use in low-and high-income countries: an overview of the peer-reviewed literature. *Journal of the royal society of medicine*, 106(4):130–142.

Bisio, I., Lavagetto, F., Marchese, M., and Sciarrone, A. (2015). A smartphone-centric platform for remote health monitoring of heart failure. *International Journal of Communication Systems*, 28(11):1753–1771.

Dehling, T., Gao, F., Schneider, S., and Sunyaev, A. (2015). Exploring the far side of mobile health: information security and privacy of mobile health apps on ios and android. *JMIR mHealth and uHealth*, 3(1):e8.

Hayes, D. F., Markus, H. S., Leslie, R. D., and Topol, E. J. (2014). Personalized medicine: risk prediction, targeted therapies and mobile health technology. *BMC medicine*, 12(1):37.

Knorr, K. and Aspinall, D. (2015). Security testing for android mhealth apps. In *Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on*, pages 1–8. IEEE.

Manjoo, F. (2015). A murky road ahead for android, despite market dominance. ISSN 0362-4331. (May 27, 2015) Retrieved Oct 17, 2016.

Melillo, P., Orrico, A., Scala, P., Crispino, F., and Pecchia, L. (2015). Cloud-based smart health monitoring system for automatic cardiovascular and fall risk assessment in hypertensive patients. *Journal of medical systems*, 39(10):1–7.

Stradolini, F., Riario, S., Boero, C., Baj-Rossi, C., Taurino, I., Surrel, G., De Micheli, G., and Carrara, S. (2015). Wireless monitoring of endogenous and exogenous biomolecules on an android interface. *IEEE Sensors Journal*, 16(9):3163–3170.

Tang, X., Hu, C., and Lin, W. (2015). Android bluetooth multi-source signal acquisition for multi-parameter health monitoring devices. In *Information and Automation, 2015 IEEE International Conference on*, pages 1790–1794. IEEE.