Bartosz PAWŁOWICZ [1], Artur STRZAŁKA [2], Mateusz TYBURA [2]
[1] DEPARTMENT OF ELECTRONIC AND COMMUNICATION SYSTEMS
[2] STUDENTS SCIENTIFIC CIRCLE OF INFORMATION TECHNOLOGY
RZESZOW UNIVERSITY OF TECHNOLOGY, 12 Powstańców Warszawy St., 35-959, Rzeszow, Poland

Privacy and security of contact data on mobile phones with Windows Phone Operating System

Abstract

In this paper we present analysis of how contact data is managed by Windows on mobile phones and our approach in making it more private. Aim of this work was to design and develop application that can store contacts locally in memory of device with installed Windows Phone 8.1 operating system. That application can be used to create, edit, and remove contacts without usage of Microsoft cloud services. Developed application also allows to send text messages, e-mails, and call to saved telephone number.

Keywords: Windows, Windows Phone, contacts, data privacy.

1. Introduction

Nowadays mobile phones are more widespread and personal than any other electronic devices. Majority of them use cloud services to store their data such as contacts, photos, documents etc. It's a challenge from privacy point of view. Many users prefer to store crucial data such as contacts only in internal memory of mobile device. Unfortunately today mobile devices often do not offer function of private store, since contacts and other data are synchronized with cloud by default. The purpose of this work was to design and develop set of applications that enable storing contact data locally and synchronizing them between devices without usage of any public or private cloud. Any system for such a thing must be fast, stable and absolutely transparent for being easy to use out of box. More complex solutions could be considered but they would be limiting possible usage to just few or more advanced users.

2. Contact data and content

Contact data in mobile phones could be only stored in SIM memory and/or internal phone memory. Unfortunately Windows Phone make saving contacts into SIM memory unavailable. There is only an option to import already stored entries. There is also possibility of synchronizing this and other data to computer or some servers, especially Google or Microsoft cloud servers (Fig. 1) which in Windows Phone 8.1 is mandatory when the user wants to use advanced features of the phone such as synchronizing e-mails between devices or install applications from the Windows Store. Also privacy settings do not allow to disable contact synchronization function. In extreme cases user who logs phone to Microsoft account to install application from store and does not use outlook e-mail might not know that contacts synchronization with cloud already occurred. In Windows 10 mobile there is a possibility of selective logon that avoid described earlier situation but similarly to Windows phone there is no tool to synchronize data locally. There are already many solutions to synchronize and manage contacts data on the market, but each has one drawback. Contacts are stored on the servers of companies. This means that company which stores data and possibly other persons such as hackers can access a data and their content, and in times when privacy is of high importance that is not particularly desirable.

More custom solutions could use custom services like Google Cloud Messaging Services [3] or systems like SugarCRM [4]. Which are still third-party components.

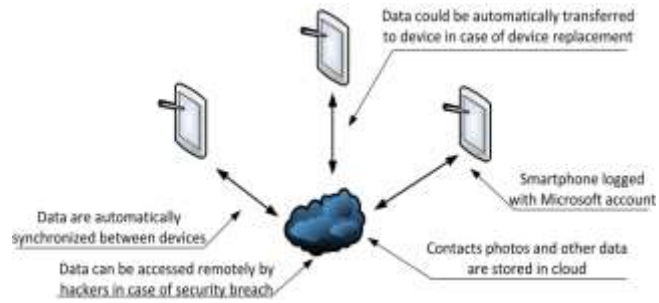

Fig. 1. Idea of data synchronization between mobile devices and cloud

Contacts provide organizing and storing information about people such as your name, e-mail addresses, phone numbers, addresses, names of employers, departments, positions, etc. By registering a Microsoft account user can get a set of tools for managing personal information.

Contacts in Windows Phone are integrated with services such as Hotmail, OneDrive or Calendar. All this data can be synchronized with Microsoft cloud applications in order to making backups and easily switching between devices with Windows Phone operating system. Thus it is quite obvious that built-in application capable of storing and managing contact data aren't taking security into account. Only locked screen with password check makes it unavailable to get in [5].

Today contacts applications gather a lot of information. Not only phone numbers with names and surnames are stored. For example "Contacts" application can also save data from Gmail, Twitter and other contact data suppliers (Fig. 2) if there is Facebook account connected with phones, we would see profile photo and last activity.

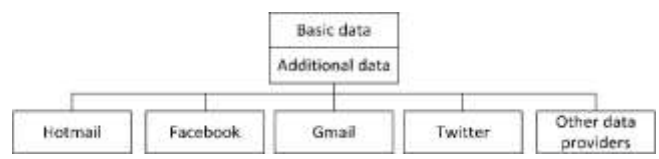

Fig. 2. Idea of internal structure of "Contacts" application data

Build in Windows Phone on the base of this fact contact data could be represented as structure build of basic and additional data. First one would simply store contact name and phone number. Second one would point to other structures with data provided from one of accounts for example Facebook account. It seems that there is too much data connected with one person in standard contact book.

3. Application concept

Application must provide easy to use solution for storing and managing data about contacts. It involved several analysis on both security and privacy. For more privacy data was slightly reduced to form defined as basic data in previous chapter. Also there is no other way for synchronization than usage of desktop application (Fig. 3).

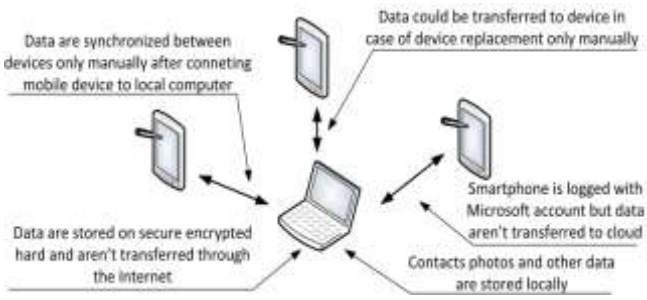

Fig. 3. Idea of data synchronization in local environment

Client-server was chosen to be used as an communication model. This way of transmission was easy to implement and use on both own machine or some third party server if necessary. It could also take advantage of HTTPS or some other secured protocol to make synchronization process slightly more secure.

Other reason for that is the fact that we couldn't simply plug phone to the USB and then copy data to the computer. There is no API for doing it by our software. If trying to do that, data would be totally exposed as it must be seen in file explorer when accessing phone files on computer. It was considered as absolutely unacceptable solution since contact data could be copied on any computer used to charge battery.

Client and server communicate with one another through the use of interfaces and communication protocols, such as TCP or UDP. For written applications, where the key element is to send and receive intact string, choice was simple - the TCP protocol that allows to establish a secure session between the devices.

The users consuming the services of companies that provide the ability to manage contacts are exposed on a review of data stored in these services by these companies. They are doing backup of contacts and store it on their servers. Users privacy is exposed and can be used e.g. by unauthorized third parties or various intelligence services and so on.

Fig. 4. User interface and basic functions of developed mobile application

The application were designed with usage of official development guides [6, 7]. They (Fig. 4 and 5) allow to access to local contacts on users computer. This may be needed while user will exchange phone and have desire to quickly restore contact information to a new device with Windows Phone on board.

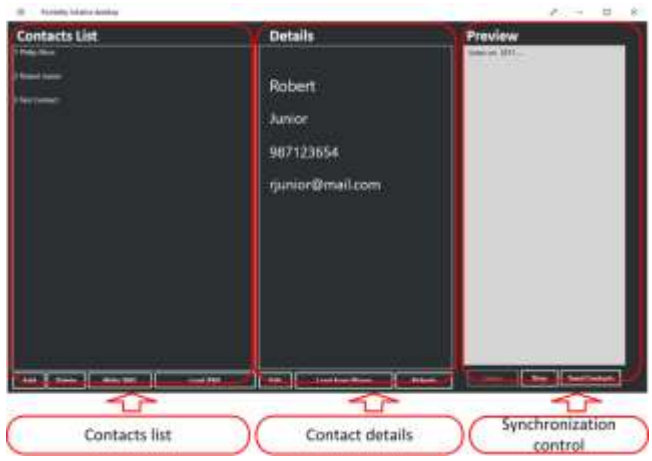

Fig. 5. User interface and basic functionality of developed computer application

The application provides methods and functions to manage a set of contacts. Its main element is a list of items with names and surnames. Clicking on a contact enables the contact details of the entity, ie. first name, last name, email address, phone number. Below the list there have been placed buttons for add, delete, and sync contacts.

In the project contacts, the list is sent between mobile device and computer in the form of a stream of XML (Fig. 6), so the transmitted data cannot be corrupted. A desktop and mobile application requires sending the entire string flow of the string of XML to be able to load it into database.

```
<?xml version="1.0" encoding="UTF-8"?> // declaration
of xml namesapce

<kontakt> //root

<imie>Artur</imie> //element

<imie/> // empty element

</kontakt>
```

Fig. 6. XML file structure

The data is stored in the form of text enclosed in tags. The left angle bracket "<" starts tag, and right ends it ">". A set of such markers creates an XML document does not have to be on file. XML makes it easy to find and fix errors. File opened in web browser will show well formatted document or error message with description of the problem. Even more there is a possibility for building special file describing how our document must be structured.

On the other hand it takes more space to store data than for example CSV file. The first line consists of elements that describe the version of XML and encoding. The root element is called contact. The root is the parent for the elements named *name* that contains the relevant data. The second part of the name is empty.
Unfortunately there is a large gap of functionality. Our approach doesn't have automated synchronization of data as in internal Microsoft solution. It is considered both good and bad, as it takes less effort to use but also makes more unconscious about what is exactly going on. On special needs this functionality could be added to application.

Also there is a lot of place for encryption algorithms and usage of system API which makes file available only for our program. Instead of simply showing data, it can be protected by showing login form on first run and any time application is hided or phone is blocked (Fig. 7).

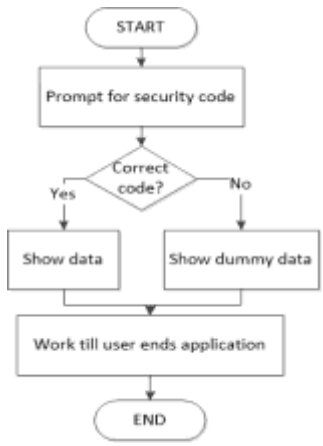

Fig. 7. Application algorithm concept

4. Cryptographic algorithms test

To make sure that data is more secured in both making it unavailable to retrieve and making it unavailable to simply read, we decided to implement and test some security improvement [8, 9, 10]. For this particular reason we made some other application which contained four own implemented ciphers. For making sure that we would only test cryptographic algorithms.

Since it is crucial to have unlimited knowledge of all encryption algorithm it was assumed to analyze their code or at least check how they work. That last assumption was done mostly because of lack of ability to fully analyze standard cryptographic algorithms which are implemented in Windows Phone, on source code and binary level

Test was done in memory because of operations such as reading and saving files could slightly increase amount of time. Text used in this test was generated using lorem ipsum generator available online at http://lipsum.com/.

Testing procedure consisted of few runs of 10-times ciphering. Then we collected minimal, maximal and average time. Making it run once could not build full image of speed because of any initialization taking place before first run. Any impact of this necessary process would just change effective time making false impression.

Caesar algorithm was the first we tested. It's minimal time value seemed to be constant in all runs. Biggest max time was measured only in first attempt undoubtedly because of initialization procedure performed on first usage of method. This longest run had taken almost 33 more time than the shortest one. Average time fluctuated from 4,43 to 1,41, which in two cases was nearest to max than min value (Tab. 1).

Tab. 1. Caesar algorithm performance

Run number	Caesar algorithm execution time, s		
	Min	***Max***	***Avg***
1	1	32,23	4,43
2	1	2,02	1,41
3	1	3,02	1,51

Next we tested Vigenere algorithm. As in Caesar algorithm it's minimal time was constant. On the other hand it value was 6 times bigger. Maximal and average time was also bigger than in first method, with one exception in first maximal time, which was 2 times smaller (Tab. 2).

Tab. 2. Vigenere algorithm performance

Run number	Vigenere algorithm execution time, s		
	Min	***Max***	***Avg***
1	6,04	16,12	7,45
2	6,04	7,05	6,14
3	6,04	7,05	6,34

On the third attempt we used Playfair cipher. This time minimal time wasn't constant, but it was just a 0,01 difference. After full initialization this algorithm worked almost 2 times faster than Vigenere, and only 3 times slowest than Caesar (Tab. 3).

Tab. 3. Playfair algorithm performance

Run number	Playfair algorithm execution time, s		
	Min	***Max***	***Avg***
1	3,01	28,20	6,34
2	3,01	4,03	3,83
3	3,02	4,03	3,93

Lastly we tested fence algorithm (Tab. 4). It was absolutely slowest of implemented by us methods of encrypting data. Minimal time was slightly higher than maximal time in other ciphers with exception for first run maximal time of Caesar and Playfair algorithms.

Tab. 4. Fence algorithm performance

Run number	Fence algorithm execution time, s		
	Min	***Max***	***Avg***
1	17,12	21,16	17,63
2	13,09	17,13	14,10
3	13,09	19,14	14,20

This measurement had shown that this method is to complicated to be used for real time communication when using mobile phones as a devices which must cipher and decipher messages. The results of conducted tests were graphically compared on Fig. 8.

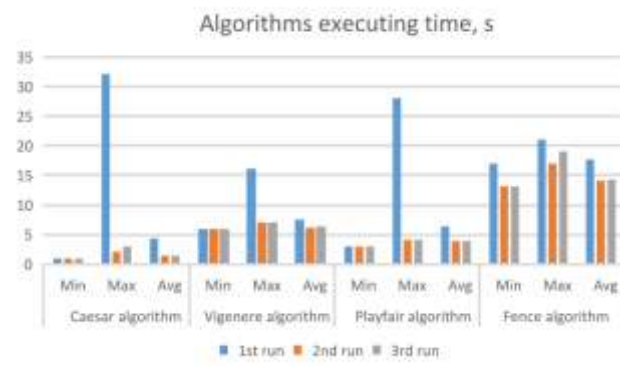

Fig. 8. Encryption algorithms execution time

5. Conclusions

Standard solutions for managing contacts data are easy to use but not aware of things such as privacy or security. Conducted tests had shown positive results of more secure approach on that particular thing but there are still many ways to improve. For instance mechanisms of synchronization could be more focused on

recognizing devices. Also this solution could be hardened by more sophisticated technique for authorization of user. Usage of both secure password, known to user and biometrics, would slightly reduce any possible interaction between application and any no authorized users.

6. References

[1] Pucci P., Manfredini F. and Tagliolato P.: Mapping Urban Practices Through Mobile Phone Data, ISBN: 978-3-319-14832-8, Springer, 2015.

[2] Androulidakis I. I.: Mobile Phone Security and Forensics A Practical Approach, ISBN: 978-1-4614-1649-4, Springer, 2012.

[3] Kedia A., Prakash A.: Data Synchronization on Android Clients, Communication Software and Networks (ICCSN), 2015.

[4] Pascaul V. S., Xhafa F.: Contact Syncrhonization for the Android Platform, P2P, Parallel, Gird, Cloud and Internet Computing (3PGCIC), 2011.

[5] Boyce J., Shapiro J. R. and Tidrow R.: Windows 8.1 Bible, ISBN 13: 9781118835319, Wiley, 2014.

[6] Windows Phone 8/8.1 specification https://dev.windowsphone.com/en-US/OEM/docs/Welcome/Windows_Phone_8.1

[7] Whitechapel A., McKenna Sean S.: Windows Phone 8 Development Internals, ISBN-13: 978-0735676237, Microsoft Press, 2013.

[8] FIPS 197 – AES specification http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[9] PKCS #1 v2.2 – RSA encryption http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsa-cryptography-standard.pdf

[10] Vaudenay S.: A Classical Introduction to Cryptography: Applications for Communications Security, ISBN 9780387258805, Springer, 2005.

Received: 02.10.2016 Paper reviewed Accepted: 02.12.2016

Krzysztof BOGUSŁAWSKI, PhD, eng.

Krzysztof Bogusławski received the MSc, eng degree in the field of electrical engineering with specialization in automation and electrical metrology from Szczecin University of Technologyand the PhD degree in the field of Telecommunications from Warsaw University of Technology. He coordinated the following projects: "Telemedicine as a eHealth in Westpomeranian Region" RPO 3.1, "Telemedicine in the Euroregion Pomerania" - Interreg IVa. His current research interests include telecommunication, computer networks and telemedicine.

e-mail: kboguslawski@wi.zut.edu.pl

Artur STRZAŁKA, eng.

Artur Strzałka completed his eng. in Computer Science at the Rzeszow University of Technology in 2016. Currently he is pursuing an MSc in Computer Systems and Networks at the Faculty Of Electrical And Computer Engineering in the same university. He is interested in Linux systems and distributed computing.

e-mail: a.strzalka93@gmail.com

Mateusz TYBURA, Msc, eng.

Mateusz Tybura has graduated from Rzeszow University of Technology in Computer Science (Msc, eng). Currently he is both working as an software developer and studying for PhD degree on the same university. He is also a member of KNEiTI scientific circle. His main interests are security, mobile technologies and artificial intelligence.

e-mail: tyburam@hotmail.com