Ivan OPIRSKYY[1], Ihor IVANCHENKO[2], Artem PLATONENKO[3],
Pavlo SKLADANNYI[3], Mariia ROSHCHUK[4]

Opiekun naukowy: Ivan OPIRSKYY

WYKORZYSTANIE TECHNOLOGII NFC DO AUTOMATYCZNEJ WYMIANY PROFILU

Streszczenie: Opracowano metodę wykorzystania technologii NFC do automatycznej replikacji profilu użytkownika. Proponowana metoda jest łatwa w użyciu, umożliwia uwierzytelnienie użytkownika za pomocą smartfona.

Słowa kluczowe: Technologia NFC, system operacyjny Android, replikacja, uwierzytelnianie, klucz sesji, Bluetooth, Mozilla Firefox.

USE OF NEAR FIELD COMMUNICATION TECHNOLOGY FOR AUTOMATED PROFILE REPLICATION

Summary: The following work contains the development of a method of using NFC technology for automated user profile replication. The proposed method is user-friendly, allows for smartphone-based user authentication.

Key words: NFC technology, Android OS, replication, authentication, session key, Bluetooth connection, Mozilla Firefox

1. Introduction

Support for user data as a means of maintaining its relevant state and the ability to access the said data from any device is the main principle of ensuring a convenient user interaction with information systems.

[1] DSc, Lviv Polytechnic National University, Associated Professor of Department of Information Protection, iopirsky@gmail.com
[2] PhD, National Aviation University, IT-security Academaic Depatment, igor-p-l@ukr.net
[3] Borys Grinchenko Kyiv University, Senior Lecturer of Department of Information and cyber security, v.buriachok@kubg.edu.ua
[4] National Aviation University, Institute of Law, Department of civil law and process, Graduate student, roshchukmv@gmail.com

The number of personal devices is no longer limited to telephones and computers only. Many people also own tablets, gaming consoles and "smart home appliances". It often becomes necessary to utilize shared devices - for example, computers in an Internet cafe or in the classroom. In order to ensure maximum user-friendliness, regardless of what device is currently being used, one needs to employ data synchronization. Synchronized data will be henceforth referred to as a user profile.
There are many technologies for synchronizing data between devices. These technologies vary depending on:

- where the synchronized profile is being stored;
- which channels are used for synchronization;
- which synchronization protocol is being utilized;
- whether any means of protection are used, and which ones in particular.

Regardless of the factors listed above, it is necessary to ensure the security of data that is being synchronized, so in place of a solution for providing data security, a relatively young technology – the NFC – has been selected.

1.1. Main part

NFC technology allows data to be transmitted at distances of up to 10 cm via a radio signal. Modern Android-based mobile phones support HCE technology, which allows to program the service handler for incoming NFC commands, while at the OS-level, it is guaranteed that the data received from the controller will be received solely by the given application. The practical performance speed of NFC (about 400 kbps) does not allow for a transmission of large volumes of data, and thus, this channel can be used for a initial key exchange in order to establish an alternative encrypted channel for direct data transfer. Bluetooth connectivity, Wi-Fi, and more can be used as examples of an alternative channel. In addition, the use of NFC technology allows for usage of the mobile phone as a key and for tracking whether it is within the field of the NFC reader. As soon as the phone is withdrawn from the field, it is possible to clear the synchronized profile from a computer. This resolves the issue of temporarily deploying a user profile, including logins, passwords, browser settings, and other data on the workstation.
Let’s determine the list of requirements for the developed method. The proposed method should: 1) use the NFC connectivity as a channel for setting up the session key; 2) use an alternative channel as a channel for data transmission, and the transmitted data must be encrypted on the session key; 3) ensure the encryption of transmitted data within the source and the receiver to establish data security; 4) initiate the removal of replicated data from the receiver upon completion of work. The scheme of the proposed method is depicted on picture no.1.

1.1.1. Prototype of a user profile replication system using NFC technology

To implement the proposed scheme, it is suggested to use a Java application that exchanges data with an Android device, which stores the encrypted user profile. It is suggested to utilize user data from the Firefox web browser as a profile. Initial setup of the session key must be conducted via the NFC technology. Java-side, a pair of keys must be generated to implement asymmetric encryption. Then, phone-side, a

session key of symmetric encryption is to be generated and transmitted Java-side via a public key. For transmission, an NFC reader ACR122U is used.
After exchanging keys, a Bluetooth channel must be established and used to transfer an encrypted profile. Java-side, the profile is to be decrypted and made accessible in the Mozilla Firefox browser.
The functional scheme of the prototype is presented on picture no.2.

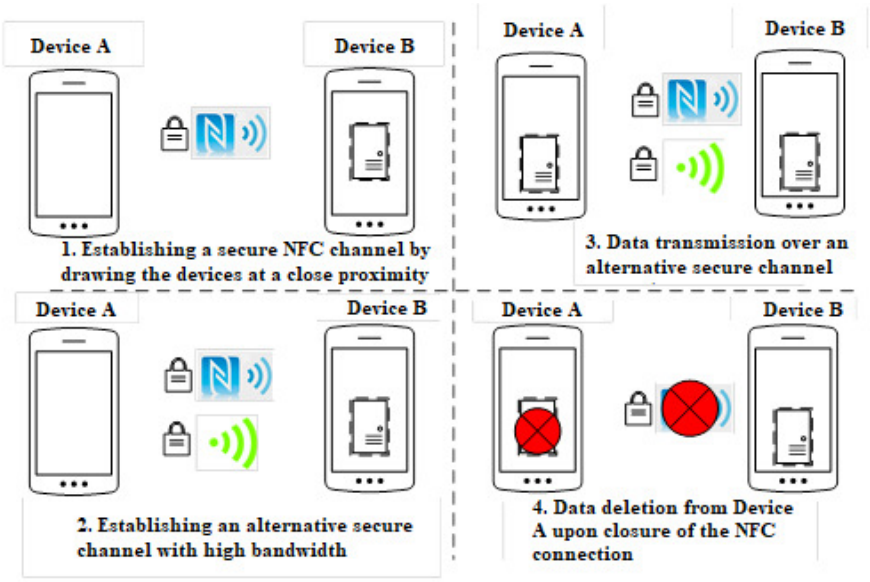

Figure 1. Replication scheme of NFC utilization for secure channel establishing

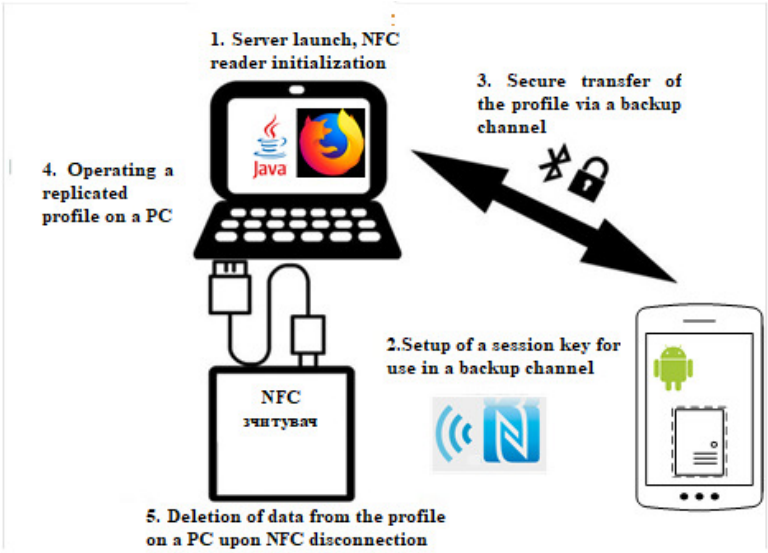

Figure 2. Functional scheme of the prototype

1.1.2. Mozilla Profile

The Mozilla Firefox browser stores user information: bookmarks, extensions, and user preferences within a unique user profile. Upon the initial launch of the browser, a profile is created by default, additional profiles can be made through the profile manager menu. All of the settings are stored in a special folder consisting of numerous files. Current settings that are related to an existing profile can be found in the folder "%APPDATA%\Mozilla\Firefox\profiles.ini". This file is used when searching for a profile during the browser launch. The file contains information in a user-friendly format that is compatible with any text editor.
In order for the browser to load our profile, we shall edit the given file programmatically, backing up its old version in advance. To do this, the Path value must be changed so that it points to a folder, inside of which the user profile will be replicated, for example: "Profiles/tempProfileFolder". Now the browser will load the profile from the specified directory upon the next launch. In addition, browser mechanisms can be used to specify a master password, which will be asked each time a site authorization attempt, using credentials stored in the browser databases, is prompted. This additional security measure is intended to help in cases of theft of key3.db and signons.json files, but it is not exhaustive. Since file data theft allows for local brute-force attacks, even though data recovery is possible, its complexity is affected by the complexity of the master password.

1.1.3. Profile storage within phone memory

For a transmission on a protected channel, the user profile is archived into a ZIP archive in order to reduce the volume of transmitted data and to use built-in ZIP integrity checks.
After transferring to a mobile device, the file is immediately stored in the internal memory of the device. Other programs will not be able to access the internal memory of our application, because the Android OS uses isolated environments for each program running on the system. To ensure additional security, data is also encrypted on a key that can not be directly requested from the application itself. The Keystone mechanism is used to store the key. Through this mechanism, the key can only be obtained if the user enters the correct access code. This approach allows to avoid risks, associated with phone loss and unencrypted file system analysis.

1.1.4. Establishing a secure channel using NFC

To ensure the safe transfer of "sensitive" data, it is suggested to use an encrypted NFC channel. Initial setup of the encrypted channel is conducted using the RSA algorithm (Rivest, Shamir, Adleman). RSA is an asymmetric encryption algorithm. Computer-side, a pair of keys – private and public – are generated. As shown on picture no.3, the public key is transferred to the mobile phone with the help of NFC.
Since the NFC channel is safe from eavesdropping and "man-in-the-middle" attacks, we can freely transfer the public key to a mobile device and subsequently set up a private key for the AES algorithm, as shown on picture no.4, as well as transfer the necessary data needed for establishing a connection to an alternative data transmission channel. Thanks to the Android Host Card Emulation technology, it's possible to create your own NFC device communications protocol. In the handler of

incoming APDU commands, both the response to the incoming command and the actions, performed by the phone when receiving the command, may be programmed. To avoid random transmissions based on relay attacks when initiating an NFC connection, the SELECT APDU command handler can added with a confirmation dialog of the user transaction start in the form of a dialog box. In addition, in the application itself allows for usage of a finite-state machine, as shown on picture no.5. Specifically, if an application received a request, prior to which there was nothing being initialized (This state may occur, for example, as a result of attacks by an intruder), then the given request is not executed, the actions are logged, and an error code is sent to the reader. This situation may occur, for example, if, after the SELECT AID command, the device receives a Bluetooth data transfer command. An error will be issued because the steps to set up the encrypted data transmission channel have not yet been made.

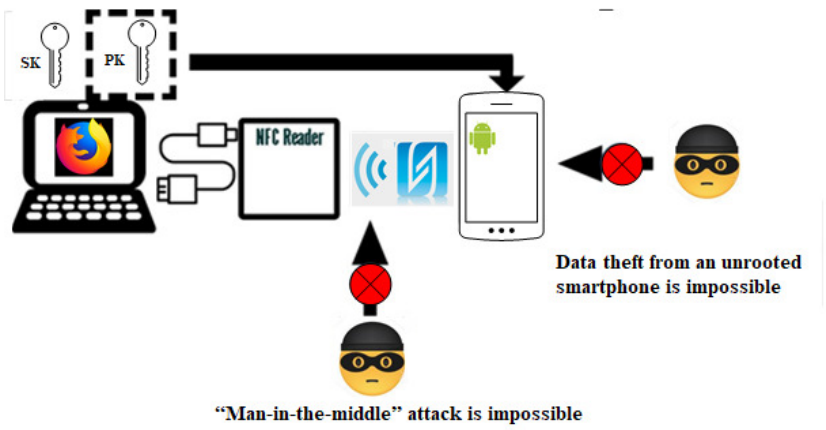

Figure 3. Establishing a public key

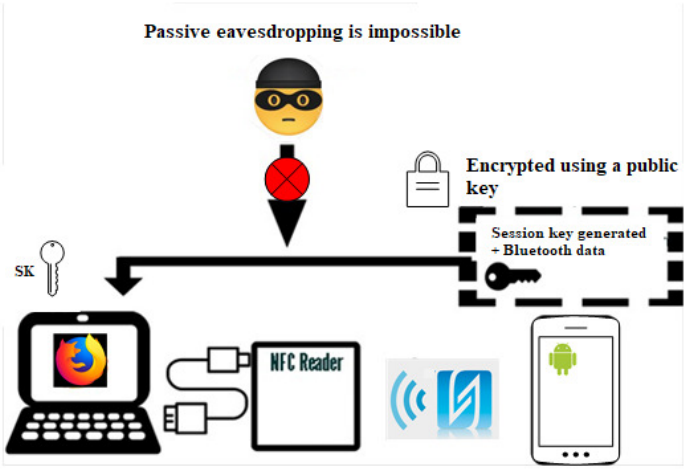

Figure 4. Establishing an NFC protocol session key

1.1.5. Data transmission using an alternative channel

The prototype uses a Bluetooth connection as an alternative channel. To provide even greater security, Bluetooth starts in direct connection mode, that is, the connection is made using a direct MAC address. Broadcast requests are not executed.
In addition, all data in the Bluetooth channel is further encrypted on a session key, which prevents the intruder from scanning the data off the channel, even if eavesdropping remains possible. The data transmission scheme for an alternative channel is shown on picture no.6.

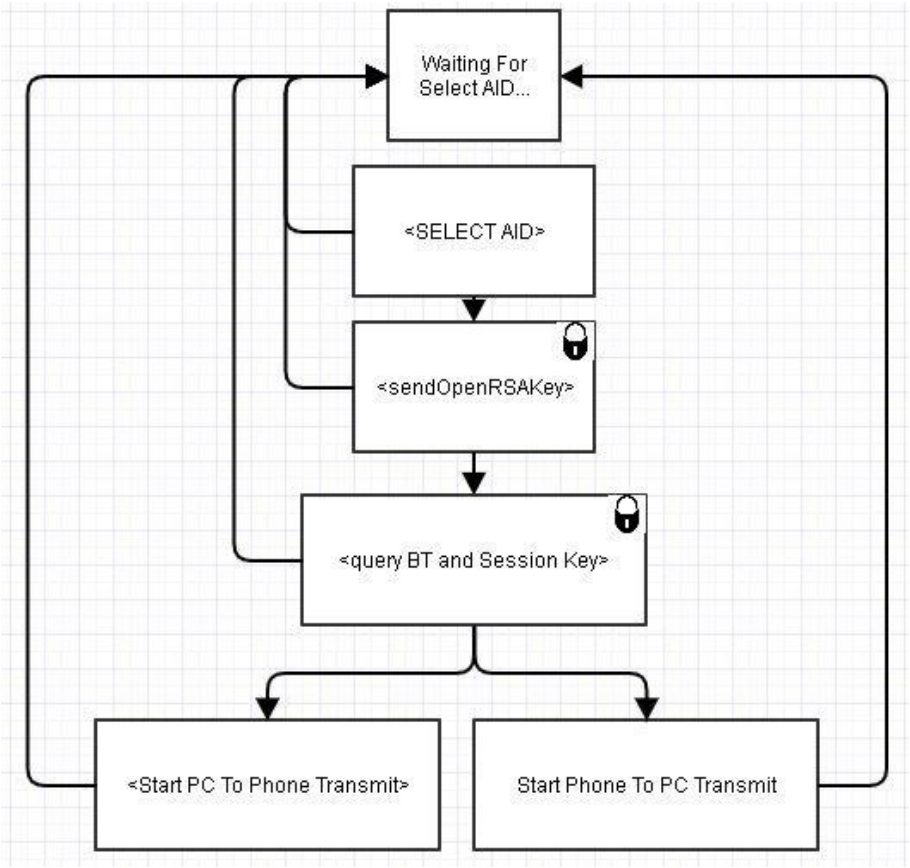

Figure 5.Final transition automata of the exchange protocol of NFC messages

After transferring the user profile to a computer, the server program determines whether the Mozilla Firefox browser is already running and halts it if necessary. Afterwards, the decoding and parsing of the transferred file take place, as well as the subsequent transfer of the given file to the profile folder, and then you can start the browser and work. The user can work with the profile as long as his mobile phone remains within the field of the NFC reader. At the moment of operation, the screen is automatically tuned to the minimum level of brightness.

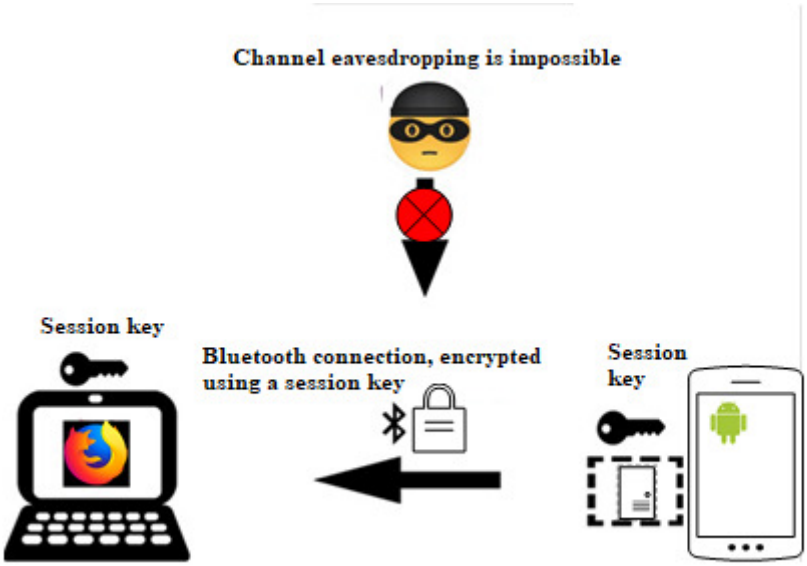

Figure.6. Data transmission scheme for an alternative channel

The phone presence search is set in the server application and is by default equivalent to about one second.
As soon as the phone disappears from the reader's field of operation, a corresponding message appears with the suggestion to bring the phone back in and continue the interaction, or exit the system and delete all data from the computer. It is worth noting that profile data, which replicates in the target system, is created as temporary files, thus, with the correct closure of the Java machine, all data will be deleted correctly.

2. Conclusion

A new method of replication, based on the utilization of the NFC technology as a session key setup channel is proposed. The proposed method is user-friendly, allows user authentication via a smartphone, and profile data, which replicates in the target system, is created as temporary files, thus, with the correct closure of the Java machine, all data will be deleted. In addition, it can be argued that the "man-in-the-middle" attack against the NFC protocol is virtually unfulfillable, and therefore it is possible to use the developed method as an additional measure during a two-factor authentication.

REFERENCES

1. Android Security Tips. URL: *https://developer.android.com/training/articles/security-tips.html/,* 01.10.2018.
2. COUTINHO S.: Introduction to the theory of numbers. RSA Algorithm = Theoretical Mathematics of the Ciphers Number and RSA Cryptography. Moscow: Postmarket, 2001. 328 p.
3. Host-based card emulation overview. URL: *https://developer.android.com/guide/topics/connectivity/nfc/hce*, 01.10.2018.
4. Host Card Emulation in Android: What does it Mean? URL: *https://www.rambus.com/blogs/host-card-emulation-in-android-what-does-it-mean/*, 02.10.2018.
5. HASELSTEINER E., BREITFUß K.:.Security in Near Field Communication (NFC). URL: *http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-20Security%20in%20NFC.pdf*, 03.10.2018.
6. SOVYN YA., NAKONECHNYY Y., OPIRSKYY I., STAKHIV M.: Analysis of Hardware Support for Cryptography in Devices of the Internet of Things. NAU: Scientific Journal "Information Security", Kyiv 24(2018)1, 36-48.