Secure Data Exchange Based on Social Networks Public Key Distribution

Krzysztof Podlaski(✉), Artur Hłobaż, and Piotr Milczarski

Faculty of Physics and Applied Informatics, University of Lodz, Łódź, Poland
{podlaski,artur.hlobaz,piotr.milczarski}@uni.lodz.pl

Abstract. The mobile devices became the most spread tools used for everyday communication. The users of mobile applications demand high level of security. All existing encryption protocols require from the users additional knowledge and resources. On the other hand the common user does not have required knowledge and skills about security. In this paper we discuss the problem of public key distribution between interested parties. We propose to use a popular social media as a channel to publish public keys. That way of keys distribution allows the owner of the key to connect easily with the desired person or institution, that is not always easy. Recognizing that the mobile devices are the main tool of communication, we present example of a mobile application that uses the proposed security method.

Keywords: Secure communication · Data encryption · Public key distribution · Mobile applications · Social networks

1 Introduction

Nowadays, people take into account security of information exchange. There are many different methods of encryption. The most spread and probably most popular are asymmetric methods based on a pair of user keys, public and private. While private key has to be kept very secret the public should be freely distributed between all interested parties and here we arrive at the big gap in used protocols. All known methodologies are very interested in securing the keys and authorization. We can sign the public key via well-known institutions (VeriSign, Comodo SSL, GlobalSign, etc.) and prove that a defined person or a company created this key. Even having a given key of a John Smith from Milwaukee how can a person be sure that this is exactly the same John Smith he knows? For many persons (even institutions) knowing their names and addresses is not enough. On the other hand if the John Smith is somebody's friend in real life he can be a "friend" in virtual one. They usually are connected via social network (Facebook, LinkedIn, etc.). The life would be much simpler if we could obtain his public key from this social network. In this paper we introduce an architecture for applications that allows sending encrypted information between two mobile devices using public key infrastructure and social media with QRcodes as a method of seamless distribution of public key.

B. Mandler et al. (Eds.) IoT 360° 2015, Part I, LNICST 169, pp. 52–63, 2016.
DOI: 10.1007/978-3-319-47063-4_5

The paper is organized as follows. In Sect. 2 we analyze the possibility of the storage of the public key with use of social media. The next Section contains requirements for QRcodes. Section 4 focuses on the used encryption method. Section 5 contains description of the proposed application architecture. At the end we present our final remarks and conclusions.

2 Public Key Distribution Using Social Media

There are many interesting methods of public key distribution. One of the well-known methods of public key distribution is usage of key servers. Conventional PKI and PGP are still hard to be used by average users [1,2]. The task to acquire valid public key of a friend is not an easy one. Nowadays, users are used to use the social networks as the environment for searching any personal information. The everyday social networks and mobile devices revolutionized ways of communication. The average user is used to integrate all mobile devices with some social medias and requires all important data to be synchronized with the device phone book. Unfortunately existing key servers are not ready to be used in such a way. Some important elements have to be taken into account during the process of public key distribution:

1. ownership of the stored public key,
2. correspondence between the owner and real party (person, company, foundation, etc.),
3. easy accessibility to all interested parties,
4. is the key still actual/valid or was revoked.

Even though we have the key from some public storage in order use it we have to be sure to whom it belongs. The name of the person or company and even address are not always enough. Analyzing presented requirements we can notice that usual PKI or PGP key distribution does not always fulfill point 2 and 4. We can try to use the webpage of a party or company, but there are often some additional problems:

– what the page address is,
– where the key is stored,
– how to obtain the key automatically.

On the other hand, the social media are the most spread and used mean of information distribution. Based on that experience there is an idea of using that medium for key distribution. First we have to analyze what kinds of information are already used in social medias. We can easily stress that on most of social portals users can store some data. The security measures used in such medias restrict that only the owner of the account can store and change this information. We can identify two types of information:

1. persistent data - like photos or images (usually more than one), web page address, email address,
2. transient data - like status, notes and memos.

The first type of information is usually stored in the users profile while the latter in some blog type medium. It is obvious that transient data is not a good candidate for our purpose. This means we should concentrate on elements that can be stored in users' profile. Moreover, it has to be noted that usually user is not allowed to customize what kind of information can be stored there.

It was already proposed in [3] to store link to our public key as one of user's web addresses. This is interesting idea however the user still has to have some special place for storing the key and social medium is used only as the information where to find the public key. Moreover, this method is easy accessible by machines while strange web addresses are not well perceived by humans.

The second very interesting place for the key repository is the user profile photo/image or gallery (if exists). This will give us a huge area for implementations, if we would be able to store the public key inside the photo gallery. Now, we will try to cover this case more carefully. We have to take into account that social media usually optimize images that often means resizing, increasing jpg compressions.

2.1 Storing Public Key in Photo Metadata

Almost all image formats allow storing some additional information in attached metadata (Exif [4], XMP [5], IPTC [6]). That would be a good place to store the public key inside the metadata of profile picture. Unfortunately, the most known social portals (Facebook, LinkedIn) erase all metadata after the upload. This means that if we upload a photo with some information added in its metadata the information would be lost and not available for others.

2.2 Storing Public Key in Photo File

There are many methods of steganography [7,8] that allow storing some information inside images. Unfortunately, these methods are very sensitive on operations like resizing and jpg optimization. All images uploaded to social medias are optimized and this operation would make impossible usage of steganography. Even hiding public key after the file closing marker would not work because all the information after the EOF (End Of File) marker is deleted by social media portal.

2.3 Storing Public Key Inside as a QRcode

QRcode [9,10] is an image that encodes some text. It is possible to store public key as QRcode. The idea of using QRcode as key exchange for secure mobile communication was presented in our previous paper [11]. That way of storing the key has some advantages:

1. QRcode does not lose information during usual image resizing,
2. QRcode does not lose information during changes of image format (.jpg, .png, .gif . . .).

Storing public key as QRcode inside user gallery agrees with all requirements. We should decide for some nomenclature of naming the file with public key QRcode. Unfortunately, we cannot store it as profile picture, most of the people prefer to use real photo in that place. On the other hand, QRcodes are so widespread that they should not be perceived as out of place in user's photo gallery.

2.4 Storing QRcode on a Given Picture

There are some possibilities to store QRcode and a given picture together. There are methods like colored QRcodes but they are not acceptable for profile picture. There are also approaches to include a QRcode inside a picture. This is however not possible for all images and small QRcodes can be lost during resizing and optimization procedures. It is possible that some encoding of QRcode in image using HSB color space would be resistant for resizing and optimization but the impact of such procedure on image itself has to be determined.

2.5 Conclusions

According to presented analyze the best choice is to store public key in form of QRcode inside user's profile on social portal. This solves easily the problem of propagation and accessibility, on the other hand keeping some information about authenticity - only user can store photos in his/her gallery and prevents phishing attacks [12,13]. The problem of revoking the old key is easy to organize in proposed manner also. Moreover, if somebody would like to narrow group of users that can view/use public key then access to galleries can be restricted to selected group of users (friends), this is possible in most of social portals.

3 Selection of QR Code Parameters

On the basis of [9–11] it was found that the best type of QR code to use for our purposes will be the version 17 (85×85). It will allow hiding a public key with the length of 2048 bits with the highest possible error correction feature - level H, approx. 30 %. In the Table 1 shown below there is short description of QRcodes capacity using different variants of QRcodes.

Table 1. QR code types and their capacity

Parameters	QR code type			
	QR code Model 1	QR code Model 2	Micro QR code	iQR code
Max Size [modules]	73×73	177×177	17×17	422×422
Max Capacity in numerals	1101	7089	35	40637
Max Capacity in alphanumeric	667	4296	21	24626
Max Capacity in binary [bytes]	458	2953	15	16928

4 Encryption Variants

Depending on the amount of data to be transferred between users, we can distinguish two possible encryption schemes [14,15] which have application in mobile implementation described in the next section: A. asymmetric cryptography, B. asymmetric cryptography together with symmetric.

Because the asymmetric cryptography is slower than the symmetric one, first variant should only be applied to transmit short information, such as chat or SMS. If the user wants to send information, he encrypts it by the receiver's public key, which he has collected earlier from the social networking site. The receiver decrypts the message with its private key known only to him. Similarly, this is done the other way (Fig. 1).

One of the problems is that this scheme above does not provide authentication/identification of the information about the sending user. To ensure the authentication of the sender, the sender should first encrypt the message with its private key, and after that encrypt it again by the public key of the receiver. This allows the receiver to be sure who is sending a message to him, because he

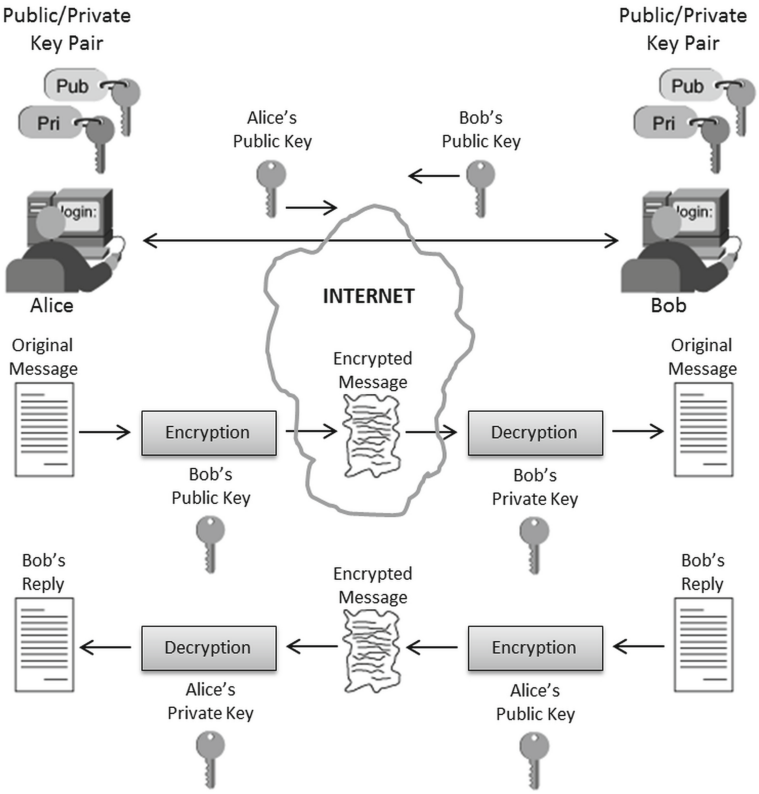

Fig. 1. Asymmetric cryptography - ensuring data integrity and confidentiality

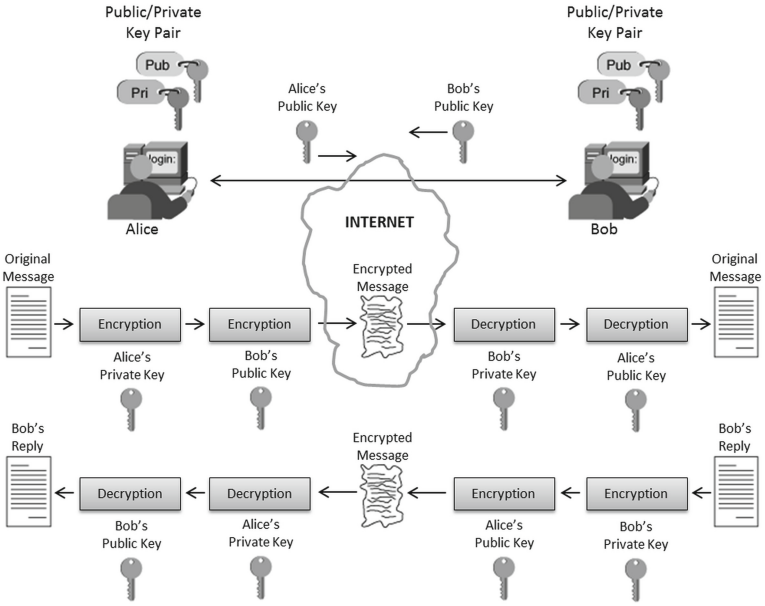

Fig. 2. Asymmetric cryptography - ensuring data integrity, confidentiality and sender authentication

will have to download the sender's public key from the social networking site in order to decrypt the information (Fig. 2).

In the case of the second encryption variant, the use of asymmetric cryptography with symmetric would allow to encrypt long information, i.e. files or stream call. In this variant, the transmission will be encrypted using symmetric cryptography. Exchange of components, which are important to establish a common one-time session key, will be done using asymmetric cryptography [16]. To establish a one-time session key, each party must first randomly generate 128 or 256-bit secret key. Secret key length depends on the used session encryption algorithm (AES-128 or AES-256). Then, the keys must be exchanged between parties using asymmetric cryptography (analogous manner as shown in Fig. 1). At this point, each party has two secret keys. To establish a common one time session key each of the sides uses XOR operation on these two secret keys (Figs. 3 and 4).

Fig. 3. The process of session key establishing

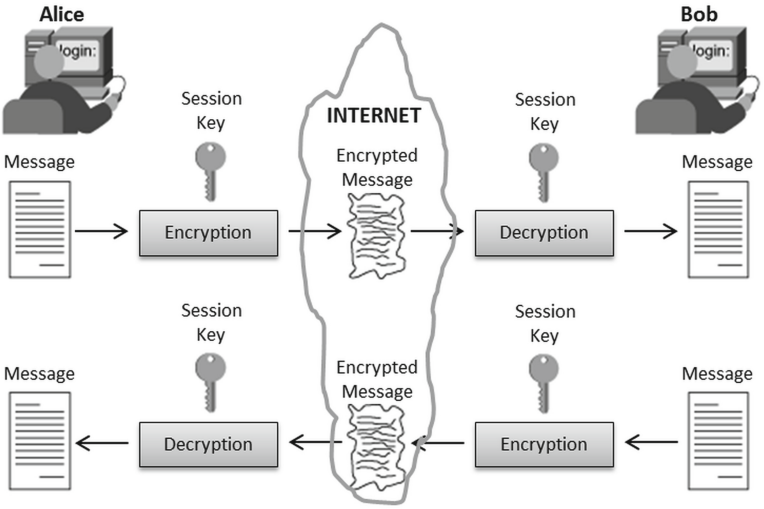

Fig. 4. Secure information exchange process

5 Analysis of Security of the Method

The proposed solution of session key establishment is different from standard methods like Diffie-Hellman algorithm. This means that no clear text information related to the key will not be possible to eavesdrop [17,18]. The attacker will be able to capture only the data already encrypted with which he will not be able to do anything.

We should analyze the impact on the presented method when somebody breaks into user's social portal account (assume it is Bob's account). Even though such possibility exists the evil party could only change the user's public key into fake one. Such action would make impossible to continue encrypted communication between Alice and Bob because Bob's original private and fake public keys would not be paired anymore. Moreover, the intruder would not be able to decrypt messages sent by Alice until the Bob's private key is compromised. In the result Bob would be informed that his social account was broken. He would probably upload original public key once again and increase security measures of his social account. This implies that the main security precautions have to be taken when implementing proposed method. This will be important to keep private key secure in mobile device application.

We can also analyse possibility of an attack of the Man in the Middle type. In order to arrange such an attack an intruder has to intercept Facebook request and supplay fake response that replace receivers public key with the public key of the attacker. This is possible with use of an appropriate fake DNS service. Next the attacker has to intercept the encrypted message before it is received by recipient, encrypt the message and resent with the appropriate public key. This means a fake BTS attack on GSM service or interception of recipients smartphone in the

presented bellow SMS communication example Sect. 6.3. For example that kind of attack is possible if the attacker uses fake Facebook ssl certificate. This can be prevented by additional checking of originality of Facebook certificate before the application downloads public key.

6 Mobile Implementation

6.1 Mobile Communication

Usually encrypted communication begins with keys exchange. In proposed method of asymmetric cryptography suites perfectly for short messages and does not need any key exchange protocols at all. Interested parties can obtain appropriate receiver key from social network and stores own private one. Therefore, it can be easily deployed for smartphones to encrypt SMS communication. The latter of proposed methods need an exchange of a session encryption key and can be used on smartphones, tablets or even laptops (devices which have screen and camera) for stream communications.

6.2 Mobile Application Requirements

In order to implement mobile application that uses the presented encryption methods with usage of social network as public key store we define the application prerequisites:

- the Internet access,
- Social Medias Network access, can upload and download files/images from them,
- save data (keys),
- can generate QRcode,
- process QRcodes,
- can capture and send SMS, or capture voice telephony agent to work with the stream voice transmission,
- can encrypt and decrypt using presented methods.

The prerequisites of the application can be widened or shortened due to the application's functionality. The general communication scheme for such application is shown diagram (Fig. 5). The idea of the method and its mobile applications for SMS messages is described below (Fig. 6). The other means of mobile communication is presented in the paper [19].

6.3 Implementation Example: Encrypted SMS Messages

The proposed encrypted method can be implemented for mobile devices as an application for secure SMS exchange. The basic idea of such application is very simple and can be described in following steps:

1. associate recipient phone number with appropriate Facebook account,

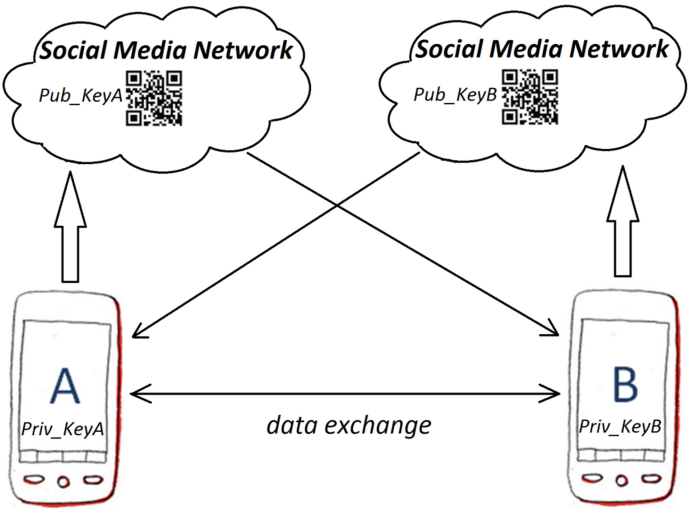

Fig. 5. General communication scheme for an application that uses proposed method

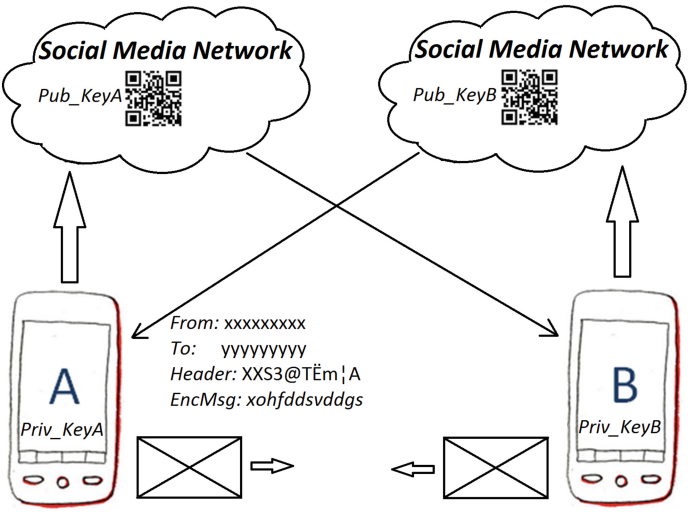

Fig. 6. Text exchange using SMS.

2. download recipient public key from Facebook gallery,
3. create and send encrypted SMS in protocol described bellow (see Fig. 6),
4. receiver's application recognizes SMS as encrypted on the base of header of the data message,
5. decrypt the received SMS.

6.4 Data Frame Format

We propose to use following Frame format to be used in different types of communication: simple data, SMS, voice calls, video, etc. In the paper we discuss the text data transmission. Below we present in Table 2 the example of the data message frame format that can be used in text messages communication, e.g. text SMS messages.

Table 2. Data message frame format.

Frame header of the message	message

In the Table 2 Frame metadata header of the message contains set number of 14 bytes:

- application ID, e.g. XXS3 – 4 bytes, where XX stands for 2 bytes special unique code,
- message type and controls, e.g. data 01000000, new key request, key received, key acknowledged, unrecognized key, etc. – 1 byte,
- timestamp of key generation of the public key (user B) in seconds – 4 bytes; we assume that users will not be able to generate keys in less than 1 s,
- version of the key used – 1 byte, e.g. RSA – 01000001 (letter A),
- message length in bytes – 4 bytes.

The proposed header can vary depending on the medium used, e.g. phone calls. In the text messaging using SMS it can be simplified by omitting the message length part of it. In the case of SMS messages the application uses simplified message header. In the header at Fig. 6 the fields stand for:

- *From: xxxxxxxx* – source phone number - provided by smartphone of user A,
- *To: yyyyyyyy* – destination phone number - provided by smartphone of user B; both fields From and To are not contained in the data message frame format,
- Header of the data message: **XXS3@TËm|A** – frame header of the SMS message, where:
  - **XXS3** – application ID (for SMS capturing by the application),
  - **@** - type of the message - data message, (seen as a text),
  - **TËm|** - timestamp as a string (1422618022 as an integer number),
  - **A** - type of encryption method, A stands for RSA,
- *EncMsg: xohfddsvddgs* – encrypted message.

In the SMS messaging the header field message length is not necessary.

After receiving the message the user B application captures the SMS (because of the XXS3 field). The application has to correlate the phone number of the user A with the Social Media profile. The user B application checks other metadata in the header. If the process of the header filtering is successful it proceeds with the message decryption and answering the message if needed. If the process of

the header filtering fails it can send control message, e.g. unrecognized key, or user B can abort the later communication with user A. It will depends on the user application setting. The advantage of such a solution is that the users can change the communication channel during the conversation.

7 Conclusions

Nowadays, there is need for secure data exchange and the mobile devices are the most spread tools used for communication. On the other hand most of users does not have enough skills to use sophisticated encryption methods and key exchange protocols. The need for secure communication and the ease of usage are very welcome by the community. The proposal of the method of secure data exchange with everyday social network as key store solves both of the problems. We analyzed in the paper the possibility of the storage of the public key with the use of social media and QRcodes. A few ways of encryption are discussed in order to allow the authentication of the sender. In order to increase efficiency of communication we propose use of symmetric cryptography with appropriate key negotiation. On the other hand the usage of images from a social network gallery makes it easy to be accepted by an ordinary mobile user. The proposed method can be implemented on mobile smartphones. Description of an example of such application was presented. The application will be presented in more detailed manner in next articles.

References

1. Ruoti, S., Kim, N., Burgon, B., van der Horst, T., Seamons, K.: Confused Johnny: when automatic encryption leads to confusion and mistakes. In: Proceedings of the Ninth Symposium on Usable Privacy and Security, pp. 5:1–5:12 (2013)
2. Sheng, S., Broderick, L., Koranda, C.A., Hyland, J.J.: Why Johnny still can't encrypt: evaluating the usability of email encryption software. In: Symposium on Usable Privacy and Security (2006)
3. Narayanan, A., Thiagarajan, N., Lakhani, M., Hamburg, M., Boneh, D.: Location privacy via private proximity testing. In: NDSS (2011)
4. Technical Standardization Committee on AV & IT Storage Systems and Equipment: Exchangeable Image File Format for Digital Still Cameras. In: Version 2.2. Japan Electronics and Information Technology Industries Association, JEITA CP-3451 (2002)
5. ISO 16684-1:2012 Graphic technology – Extensible metadata platform (XMP) specification
6. IPTC Standard Photo Metadata IPTC Core 1.2. International Press Telecommunications Council (2015)
7. Anderson, R., Petitcolas, F.: On the limits of steganography. IEEE J. Sel. Areas Commun. **16**, 474–481 (1998)
8. Kessler, G.C., Chet, H.: An overview of steganography. Adv. Comput. **83**(1), 51–107 (2011)
9. BS ISO/IEC 18004:2006. Information technology. Automatic identification and data capture techniques. QR Code 2005 bar code symbology specification

10. http://www.qrcode.com/en/codes/
11. Hłobaż, A., Podlaski, K., Milczarski, P.: Applications of QR codes in secure mobile data exchange. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2014. CCIS, vol. 431, pp. 277–286. Springer, Heidelberg (2014). doi:10.1007/978-3-319-07941-7_28
12. Vidas, T., Owusu, E., Wang, S., Zeng, C., Cranor, L.F., Christin, N.: QRishing: the susceptibility of smartphone users to QR code phishing attacks. In: Adams, A.A., Brenner, M., Smith, M. (eds.) FC 2013. LNCS, vol. 7862, pp. 52–69. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41320-9_4
13. Tamir, C.: AVG (AU/NZ) Cautions: Beware of Malicious QR Codes. PCWorld (2011)
14. Ferguson, N., Schneier, B., Kohno, T.: Cryptography Engineering: Design Principles and Practical Applications. Wiley, New York (2010)
15. Gollmann, D.: Computer Security, 2nd edn. Wiley, New York (2006)
16. Stallings, W.: Cryptography and Network Security: Principles and Practice. Prentice Hall, Upper Saddle River (2010)
17. Nikiforakis, N., Meert, W., Younan, Y., Johns, M., Joosen, W.: Sessionshield: lightweight protection against session hijacking. In: Erlingsson, Ú., Wieringa, R., Zannone, N. (eds.) ESSoS 2011. LNCS, vol. 6542, pp. 87–100. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19125-1_7
18. Adid, B.: Sessionlock: securing web sessions against eavesdropping. In: Proceedings of the 17th International Conference on World Wide Web, pp. 517–524 (2008)
19. Milczarski, P., Podlaski, K., Hłobaż, A.: Applications of Secure Data Exchange Method Using Social Media to Distribute Public Keys. In: Gaj, P., Kwiecień, A., Stera, P. (eds.) CN 2015. CCIS, vol. 522, pp. 389–399. Springer, Heidelberg (2015). doi:10.1007/978-3-319-19419-6_37