How Safe Is Safety Number? A User Study on SIGNAL's Fingerprint and Safety Number Methods for Public Key Verification

Kemal Bicakci[1,3(✉)], Enes Altuncu[1], Muhammet Sakir Sahkulubey[1], Hakan Ezgi Kiziloz[2], and Yusuf Uzunay[3]

[1] TOBB University of Economics and Technology, Ankara, Turkey
{bicakci,ealtuncu,msahkulubey}@etu.edu.tr
[2] University of Turkish Aeronautical Association, Ankara, Turkey
hakan.kiziloz@ceng.thk.edu.tr
[3] Securify Information Tech. and Security Training Consulting Ltd., Ankara, Turkey
{kemal.bicakci,yusuf.uzunay}@securify.com.tr

Abstract. Communication security has become an indispensable demand of smartphone users. End-to-end encryption is the key factor for providing communication security, which mainly relies on public key cryptography. The main and unresolved issue for public key cryptography is to correctly match a public key with its owner. Failing to do so could lead to man-in-the-middle attacks. Different public key verification methods have been proposed in the literature. The methods which are based on verification by the users themselves are preferable with respect to cost and deployability than the methods such as digital certificates that involve the use of trusted third parties. One of these methods, fingerprinting was recently replaced by a method called safety number in the open source messaging application, SIGNAL. The developers of SIGNAL claimed this change would bring usability and security advantages however no formal user study was conducted supporting this claim. In this study, we compare the usability and security aspects of these two methods with a user study on 42 participants. The results indicate with significance that the safety number method leads to more successful results in less time for public key verification as compared to the fingerprint method.

Keywords: Public key verification · Safety number · Fingerprint Usability · SIGNAL

1 Introduction

Secure communication over the Internet has become a crucial need for all of us, today. End-to-end encryption is the name of the tool and the technology that ensures encryption is performed with the keys stored only at the users' devices,

L. Chen et al. (Eds.): ISC 2018, LNCS 11060, pp. 85–98, 2018.
https://doi.org/10.1007/978-3-319-99136-8_5

hence the communication providers and other third parties are not able to read the message content. However, first generation of end-to-end encryption tools such as PGP have not become widespread mainly because they are not easy to use [3,4,6,9].

Since the first days of end-to-end encryption, two important aspects in secure messaging landscape have changed: (i) with smartphones taking place of personal computers (PCs), communication has mostly been through mobile devices, and (ii) the awareness about privacy and security has increased.

The new generation of messaging applications falls into two categories from security point of view; applications that provide encryption between the user and a server and applications that provide end-to-end encryption. Applications in the first category allow the service provider and others to read the messages being sent and received hence offer very little for those caring their security and privacy [7].

The requirements for a mobile messaging application targeting mainly consumer market to provide end-to-end encryption is minimum. Unlike some enterprise settings, digital certificates and trusted third parties are not involved. As soon as the application is downloaded to the smartphone, and registration is made via the mobile number, required keys for end-to-end encryption are generated. A central database stores identities of the registered users, and shares corresponding identities to the users that want to communicate with others. This system is not free of risks. For instance, if the central database is intentionally manipulated by the system administrator or manipulated as the outcome of an attack and user identities and/or their public keys are changed, man-in-the-middle attacks are possible. As a result, the ability to verify communicating parties via their public keys becomes essential for a secure communication.

Public key verification is presented through various user interfaces. Recently, messaging application SIGNAL has made a change and began using a method called safety numbers instead of fingerprints. In our work, our aim is to analyze security and usability aspects of this new public key verification method and compare it against fingerprint method. For this purpose, we design and conduct a user study using previous and current versions of SIGNAL on Android smartphones. SIGNAL Android application is chosen in our work for several reasons. First, its open source nature allows us to set up our own environment for the user study (this is especially important to test the older version). Second, it is one of the few messaging applications which seem to realize the importance of public key verification problem (the rationale behind the transition to safety number method - happened in November 2016 [5] - is well-documented in their official blog, though it lacks any formal supporting arguments). Finally, WhatsApp, another instant messaging application having more than a billion users, uses the SIGNAL protocol for end-to-end encryption [2].

Its leadership as a privacy advocate and general public acceptance of security and trust offered by SIGNAL hints us that the safety number method becomes more and more widely used in applications. Therefore, we strongly believe a timely user study to investigate the security and usability of this public key

verification method fills an important hole. To our best knowledge, our study is the first in evaluating safety number method and comparing it with fingerprint method with a formal user study. For this purpose, we devise a user study in a lab environment with 42 users. The obtained results indicate that public key verification via safety number method is indeed more advantageous than the public key verification via fingerprint method with respect to success rate and verification time.

The rest of the paper is organized as follows. We briefly give background information and present related work in the following section, Sect. 2. In Sect. 3, we describe the user study in detail and define our hypotheses for the study. After presenting the obtained results, we discuss them together with statistical analysis in Sect. 4. Finally, we finish up the paper with concluding remarks in Sect. 5.

2 Background and Related Work

In this section, we briefly describe the session establishment and maintaining protocol used in SIGNAL and the public key verification methods used in this study; safety number and fingerprint method. We also present the concerned threat model and summarize the earlier work.

2.1 SIGNAL Protocol

The SIGNAL protocol consists of three main parts: registration to the central server, establishment of the messaging session, and the actual message exchange (see Fig. 1).

When a user registers with their mobile number on the SIGNAL server, the key sets are generated in the background. The generated private key is stored at the mobile device, and never shared with any other party (including the central server). When the user sends or receives a message request, the corresponding public keys required to establish a mutual session is downloaded from the central server during the initial setup. Session keys remain same as long as the application is not removed from the device or sessions are not re-established. Hence, they generally last for a long time, e.g. weeks, months or years. Once the messaging session is established, identity keys of the communicating contacts are stored

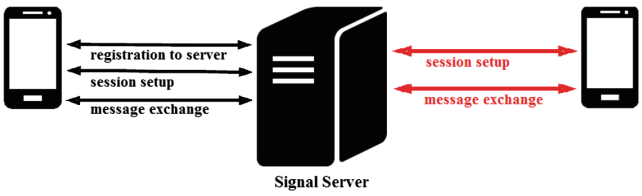

Fig. 1. A basic demonstration of how SIGNAL protocol works.

in the application. These keys can be accessed within the application. The public key verification process is simply determining whether these keys belong to the correct person or not. Normally, users are expected to perform verification upon session establishment or a genuine phone change of the communicating party.

The SIGNAL server is responsible for two tasks: relaying the messages to respective users and distributing public keys. Neither encryption nor decryption is performed on the server. When a user wants to establish a message session with another, (s)he requests their key set from the central server along with the identity key of that user. With the obtained key set, the user generates a symmetric key that is used to encrypt and decrypt messages. The same process is performed at the other side and the same symmetric key is generated. Afterwards, secure messaging takes place using this symmetric key. Both parties keep the identity key of their conversation partner.

2.2 Public Key Verification with Fingerprints

In the fingerprint method, each registered user has a fingerprint that is generated from her public key. This fingerprint is represented as a hexadecimal value consisting of 66 characters. For verification, the user must access the verification page first. This page consists of fingerprints of both users (see Fig. 2). Successful

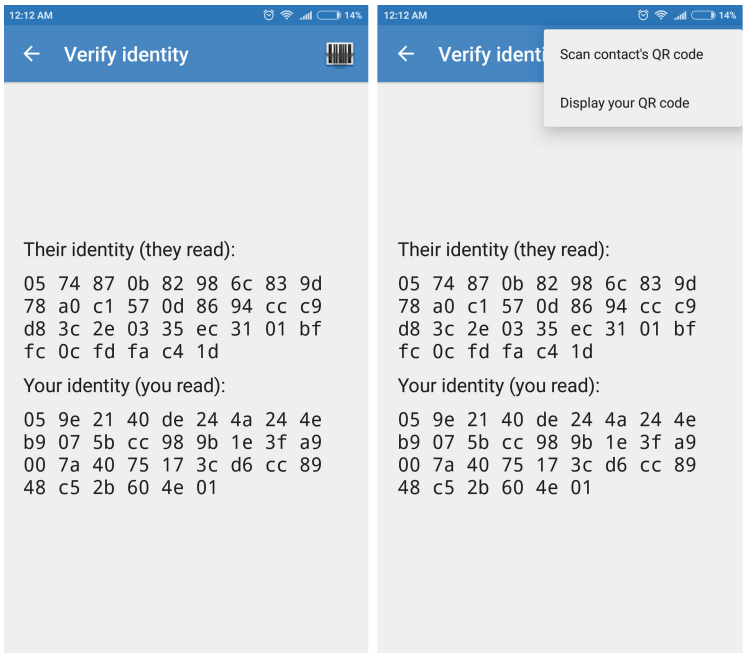

Fig. 2. Fingerprints on the verification page with a barcode symbol on top right. The displayed menu when the barcode symbol is touched is shown on the right side.

verification requires both users to ensure her own fingerprint is identical to the fingerprint on the corresponding part of the conversation partner's device.

It is also possible to perform the comparison via QR codes. For this purpose, one user must display their QR code via the "Display your QR code" button, while the partner must scan the displayed QR code via the "Scan contact's QR code" button. Similar to manual verification, this process must be repeated for the other device. Examples for matched (left) and unmatched (right) QR code scan results are given in Fig. 3.

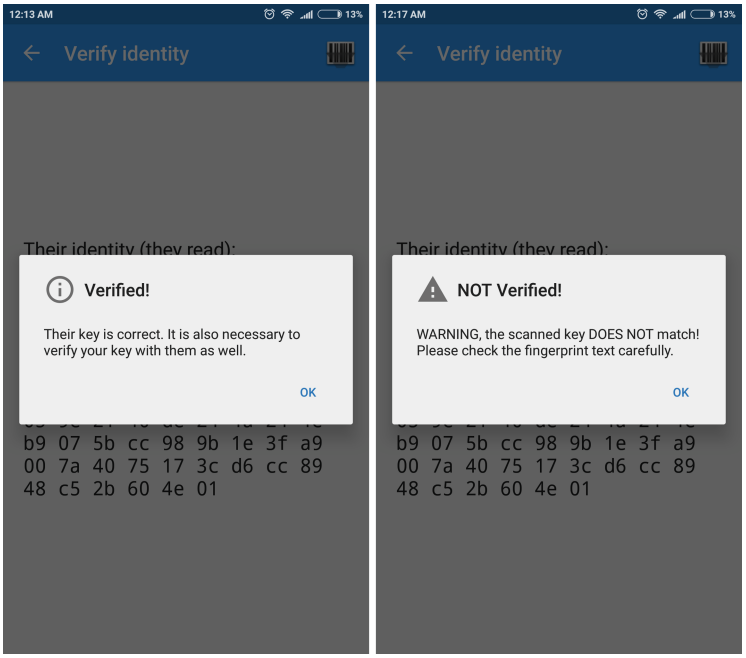

Fig. 3. QR scan results in the fingerprint method.

2.3 Public Key Verification with Safety Numbers

Unlike the fingerprint method, verification requires comparison of a single safety number that is derived from the public keys of the communicating parties in the safety number method. The safety number is specific to the session; that is, each session has its own different safety number. This safety number consists of 60 characters, represented as groups of 5-digit integers (see Fig. 4).

Upon reaching the verification page, users have two options to verify the safety number: they may either compare the 5-digit integers represented on their mobile devices by themselves, or they can use the QR scanning option similar to the previous method. Since the safety number is same in both devices, one verification is adequate. Examples for matched (left) and unmatched (right) QR code scan results are given in Fig. 5.

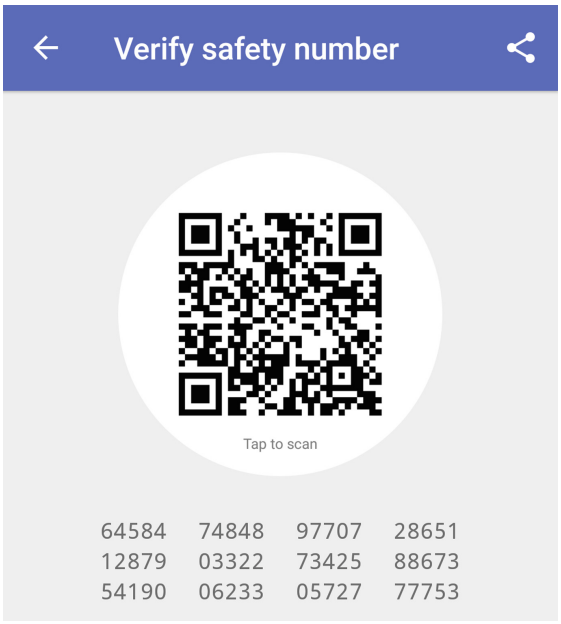

Fig. 4. Safety number and its QR code on the verification page.

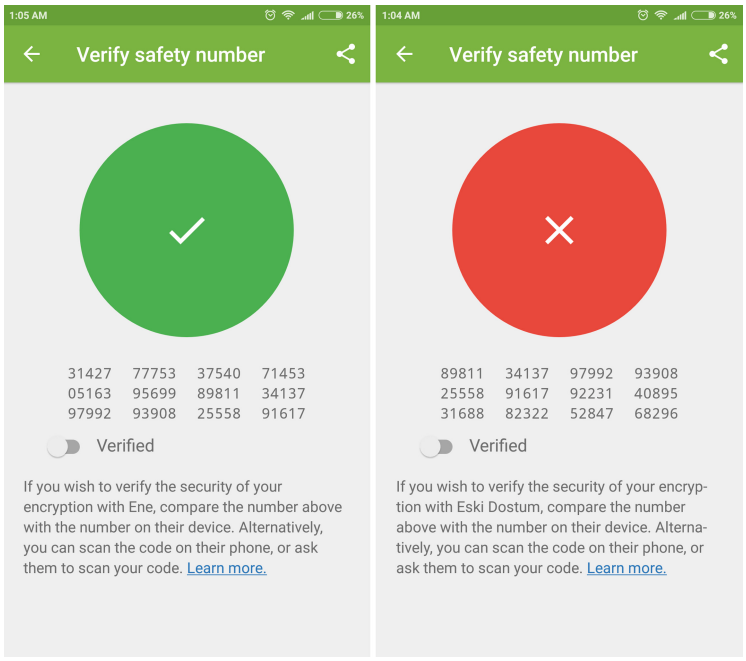

Fig. 5. QR scan results in the safety number method.

2.4 Threat Model

As mentioned before, secret session keys are generated and stored at the end devices (smartphones). The cryptographic algorithms and protocols are assumed to be secure hence it is impossible for a third party to generate the session key. However, the SIGNAL central database may be manipulated by a system administrator or by an external attacker, and hence, attacks against the session (either during the initial setup or against the already established session) may become possible.

Our work focuses on attacks against the established session. Specifically, we envision a scenario in which two users have already established a messaging session and then an attacker sends his key set to the users via the central database. In such a case, SIGNAL warns its users and suggests them to perform public key verification. We simulate such an attack scenario in our user study and investigate the effectiveness of implemented measures against this kind of threat.

2.5 Related Work

Public key verification is not a new problem, however expecting it to be performed successfully by novice smart phone users who only want to chat with their friends may be considered a more challenging problem. Having said that, the results of Whitten and Tygar's user study measuring the usability of PGP to send and receive end-to-end encrypted e-mails showed that the problem is a difficult one even for the experienced users [9]. Follow-up studies has shown that email settings are especially problematic with respect to public key verification e.g., [3,4].

For secure messaging settings, the work by Schroder et al. presented a user study for the earlier fingerprint method of the SIGNAL messaging application for public key verification [7]. 21 of the 28 users who participated in this study failed to compare fingerprints.

In a recent work by Tan et al. the authors compared various public key verification interfaces [8]. 661 participants attended the large-scale user study which involve many different fingerprint representations. One of their conclusions is that all the representations and configurations they experimented with exhibited higher rates of successful attack than seems desirable for high-risk situations.

3 Methodology

In this section, we describe the methodology of our user study and the hypotheses set prior to the study.

3.1 Design

The user study was carried out at the Information Security Lab of TOBB University of Economics and Technology. A total of 42 users were initially split into

two similar groups of 21 users, and each group was tested with a different version of the SIGNAL application. Apart from the application version, which uses a different method for public key verification, all other factors were kept same for each group. Participants were specifically asked to think aloud during the study to understand their train of thought at every step. Participants were asked to answer some questions on their demographic status and their overall opinion about the study during and after the study.

3.2 Scenario

At the beginning of the study, we briefly informed the participants about the study, and we gave them three pieces of information. The given information consists of postal address, bank account and credit card information. We asked them to send this information, one by one, to the operator through SIGNAL application that is preinstalled on a smartphone. We told participants to assume that given information were their personal information, and the operator was a close relative.

The rest of the scenario can be summarized as follows:

i. Initially, at the preliminary stage, we asked the user to open the SIGNAL application and send the address part to the operator. At the end of this stage, a session has been established between two users (the participant and the operator).
ii. Then, before moving onto the next stage, the user is asked to answer demographic questions. While the user was answering these questions, we moved the operator's sim card into another smartphone, and registered to the system with the same mobile phone number. Registering to the application from a different smartphone causes the change of public key for this mobile phone number, and hence, existing key set for the session will no longer match. This action basically simulates a man-in-the-middle attack, since a successful attack would also change the public key of the user. We note that the participants were kept busy with demographic questions and they did not see the actions taken by the operator.
iii. After answering demographic questions, we asked the user to send her bank account number to the operator. Users in different groups were encountered with different types of error messages at this point. The users of the old version of the application (which uses the fingerprint method) were encountered with an error message with a red exclamation mark (see Fig. 6), while users of the new version of the application (which uses safety number method) were warned with a mere text (Fig. 7). Both versions of the application allow users to access the public key verification page with a single tap at the prompts. At the end of this stage, users who wanted to continue with the next step without entering the public key verification page were asked to verify the identity of their conversation partner within the application.

iv. Finally, we asked the user to send their credit card information to the operator over the SIGNAL application. Since the public key does not match, the correct behavior for a user would be to discontinue the study and not to send the sensitive information.

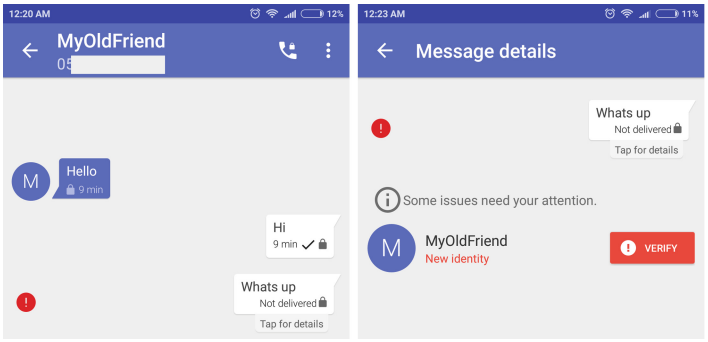

Fig. 6. Left: "Not Delivered" error. Right: Warning that the change of credentials (fingerprint method). (Color figure online)

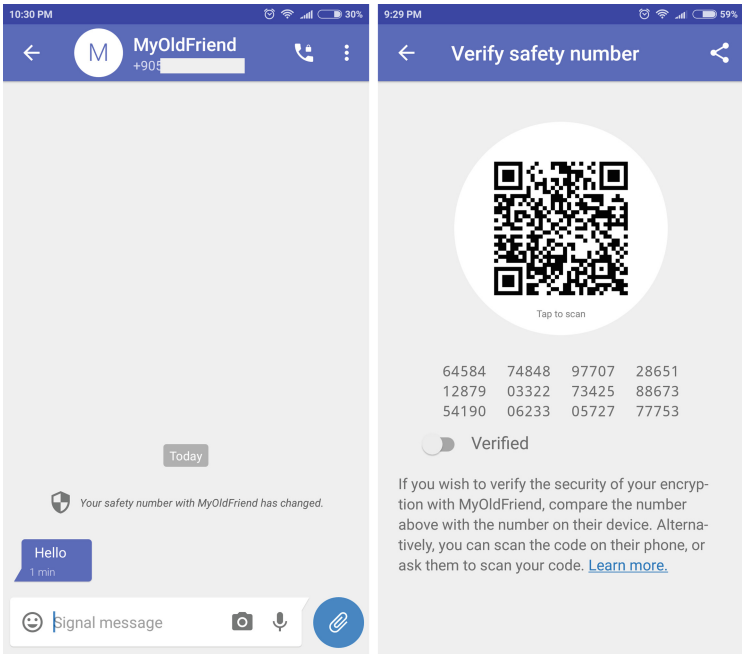

Fig. 7. Left: Warning of safety number change. Right: Safety number verification page (safety number method).

Participants were not forced to send any information to the operator at any stage, and they were informed that they could communicate with the operator face to face whenever they needed. Apart from that, the operator sat at the remote corner of the laboratory and did not play an active role at any stage. Also, he did not inform the user about verification process. After the verification process, we asked the users to answer a short questionnaire about the study and their understandings of man-in-the-middle attack.

3.3 Study Environment

We used three smartphones (Android 6.0) and one computer in our study. We used the computer for testing the old version of SIGNAL (version 3.13.0) with two purposes: as the central database server, and also as a wireless access point providing Internet to smartphones. For testing the new version of SIGNAL, we used the current SIGNAL application (version 4.11.5) that was readily available on the app store. The user dealt with only one smartphone, while remaining two smartphones were used by the operator. All the devices were fully restored to their original state for each participant.

There were 42 users aged between 18 and 25 participated in this user study. All participants were TOBB University of Economics and Technology students who were enrolled to the "Introduction to Computer Science" course in Computer Engineering department, and they were rewarded with extra credits in the course for participating. Among participants, 18 (43%) of them were female and remaining 24 participants (57%) were male. Participants were split into two similar, equally sized groups. The study lasted about 15 to 25 min for each participant.

All participants stated that they were actively using WhatsApp (42) as instant messaging application, while SnapChat (28), Skype (22), Facebook Messenger (14), Telegram (5), Discord (2), Viber (1), Bip (1), Signal (1) were also used.

Finally, participants expressed their knowledge level on information security as follows: 31 of them (74%) chose none or very low, 7 of them (17%) chose medium, and 4 of them (9%) chose high. None of the 42 participants expressed themselves as an expert. Among all, 5 participants (12%) have expressed they had prior information about man-in-the-middle attacks.

3.4 Hypotheses

On our self-trials, we encountered a system behavior in the previous version of the SIGNAL application which did not allow sending new messages after the conversation partner's public key changes until verification is completed; whereas, current version warns the user about a possible security problem, yet, allows communication. It has been shown with user studies that, active and obstructive actions, such as blocking communication, does not improve security [1]. Still we wanted to re-evaluate the effects of active (blocking communication)

and passive (allowing communication with a warning) actions on directing users to public key verification process in our study.

Since the main goal of the study is to compare the effectiveness of two methods for public key verification, fingerprint and safety number, we compare user success rates and time for comparison in each method. We expect to have higher success rates and lower comparison times for the safety number method, since comparing 12 blocks of 5-digit integers seems easier than comparing two blocks of 66 hexadecimal characters.

As a result, we specify our null hypotheses as follows:

H1 In terms of directing users to the public key verification process, there is no difference between allowing or blocking sending new messages after conversation partner's public key changes.
H2 There is no difference between the safety number and the fingerprint methods in terms of key verification success.
H3 There is no difference between the safety number and the fingerprint methods in terms of public key comparison time.

4 Results and Analysis

As described in Subsect. 3.2, users were initially asked to send the provided address information to the operator. All participants completed this preliminary task successfully. After sending the address, we asked the participants to answer some demographic questions. As they were answering the questions, we transported the sim card of the operator into another smartphone. This action changed the operator's public keys for simulating the man-in-the-middle attack. Next, users were asked to send their bank account information to the operator. Different versions of the SIGNAL application behaved differently at this point. Older version, which uses the fingerprint method for public key verification, disallows its users to send new messages until public key verification is complete; on the other hand, the current version that uses the safety number method warns the user with a passive message. We observed participants in both groups, and noticed that 4 out of 21 participants (19%) accessed the verification page through the provided link in the older version (v3.13.0), whereas only 1 participant accessed it using the provided link in the current version (v4.11.5) (see Fig. 8). The chi square test for independence suggests no significant difference in effects of employing either a blocking or non-blocking warning message on security ($\chi^2 = 2.04, p = 0.15$), resulting in insufficient evidence to reject the null hypothesis H1.

Participants, who did not access the public key verification page by themselves were directed to it, and the study continued with final phase, where they were asked to send their credit card information. Since the public key does not match, we expected the users to end the communication without sending the credit card number. As a result, we define a successful verification as the combination of realizing the mismatch and not sending the credit card information. We believe that, this definition is reasonable because participants stated that

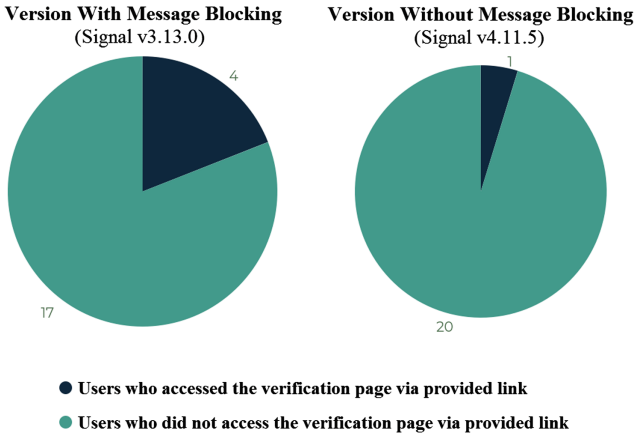

Fig. 8. Performances of the SIGNAL applications used in the study, on directing users to the verification page.

they all use the WhatsApp application in their daily life, and WhatsApp implements the SIGNAL protocol. They might have to deal with such situation by themselves in a real-life scenario. According to this metric, 9 out of 21 participants (43%) were considered successful in the public key verification via safety number method; whereas only 3 of the 21 participants (14%) were successful in the public key verification via fingerprint method (see Fig. 9). The chi square test for independence suggests marginally significant difference in choice of public key verification methods ($\chi^2 = 4.2, p = 0.04$). As a result, we reject the null hypothesis H2 in favor of the safety number method.

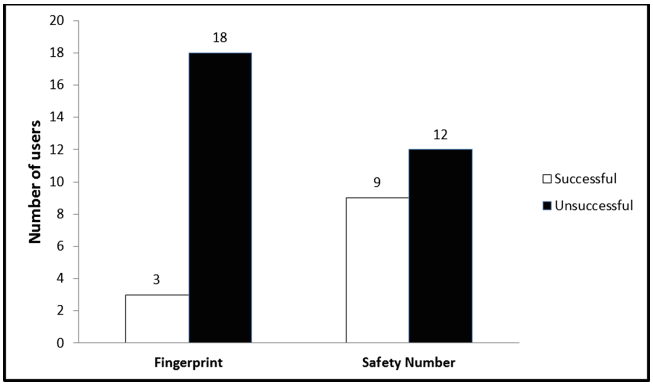

Fig. 9. Verification success rates of users.

Finally, average public key verification times for the safety number method and the fingerprint method are 82 s and 130 s, respectively (Fig. 10).

According to the Mann-Whitney-U test, a between-group median comparison test for not normally distributed data, the difference is marginally significant ($W = 58, p = 0.04$), and hence, we reject the null hypothesis H3 favoring the safety number method.

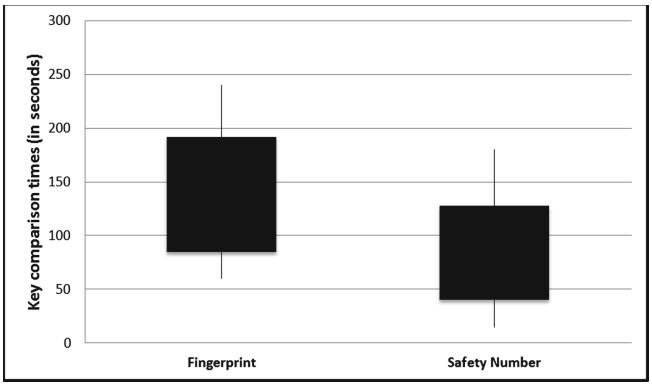

Fig. 10. Average public key comparison times of users.

To sum up, there was not enough evidence to reject the null hypothesis H1: a blocking warning message does not increase security as compared to non-blocking warning message. On the other hand, hypotheses H2 and H3 were rejected favoring the safety number method: the safety number method leads to more successful verification in less amount of time, when compared to the fingerprint method. Lastly, we acknowledge that involving more participants that represent the general user profile better could help us draw more significant results.

5 Conclusion

End-to-end encryption has become a necessity for secure communication. Smartphone communication applications provide high availability and mobility to its users; hence, their market share increase day by day. On the other hand, they must compete each other to satisfy their users' security demands.

WhatsApp is a popular communication application with more than a billion users, and it uses the SIGNAL protocol for end-to-end encryption. The SIGNAL protocol has recently changed an important feature: its public key verification method. In its older versions, it used the fingerprint method for public key verification, whereas, the safety number method for public key verification is now utilized in its current version.

In this study, we presented a user study which evaluates and compares the security and usability aspects of the fingerprint and the safety number methods

used in older and current versions of the SIGNAL messaging application, respectively. The results of our user study indicate that users achieve more success and spend less time with the safety number method as compared to the fingerprint method for public key verification. Although our results indicate a significant improvement with the new safety number method, we argue that the obtained results does not reflect yet that the problem has been solved i.e., still majority of users could not successfully perform public key verification even with the safety number method before transmitting a sensitive message. Hence, we urge usable security researchers to continue working on the public key verification problem since there is still an obvious room for more improvement.

References

1. Bicakci, K., Atalay, N.B., Kiziloz, H.E.: Johnny in internet café: user study and exploration of password autocomplete in web browsers. In: Proceedings of the 7th ACM Workshop on Digital Identity Management, pp. 33–42. ACM (2011)
2. Budington, B.: Whatsapp rolls out end-to-end encryption to its over one billion users (2016). https://www.eff.org/deeplinks/2016/04/whatsapp-rolls-out-end-end-encryption-its-1bn-users. Accessed 20 Apr 2018
3. Fry, A., Chiasson, S., Somayaji, A.: Not sealed but delivered: the (un) usability of s/mime today. In: Annual Symposium on Information Assurance and Secure Knowledge Management (ASIA 2012), Albany, NY (2012)
4. Garfinkel, S.L., Margrave, D., Schiller, J.I., Nordlander, E., Miller, R.C.: How to make secure email easier to use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 701–710. ACM (2005)
5. Marlinspike, M.: Safety number updates (2016). https://signal.org/blog/safety-number-updates/. Accessed 20 Apr 2018
6. Renaud, K., Volkamer, M., Renkema-Padmos, A.: Why doesn't Jane protect her privacy? In: De Cristofaro, E., Murdoch, S.J. (eds.) PETS 2014. LNCS, vol. 8555, pp. 244–262. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08506-7_13
7. Schröder, S., Huber, M., Wind, D., Rottermanner, C.: When signal hits the fan: on the usability and security of state-of-the-art secure mobile messaging. In: First European Workshop on Usable Security (EuroUSEC 2016) (2016)
8. Tan, J., Bauer, L., Bonneau, J., Cranor, L.F., Thomas, J., Ur, B.: Can unicorns help users compare crypto key fingerprints? In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3787–3798. ACM (2017)
9. Whitten, A., Tygar, J.D.: Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In: USENIX Security Symposium, vol. 348 (1999)