MA-INF 3319
Usable Security and Privacy
Lab Report

Matanat Ahmadova

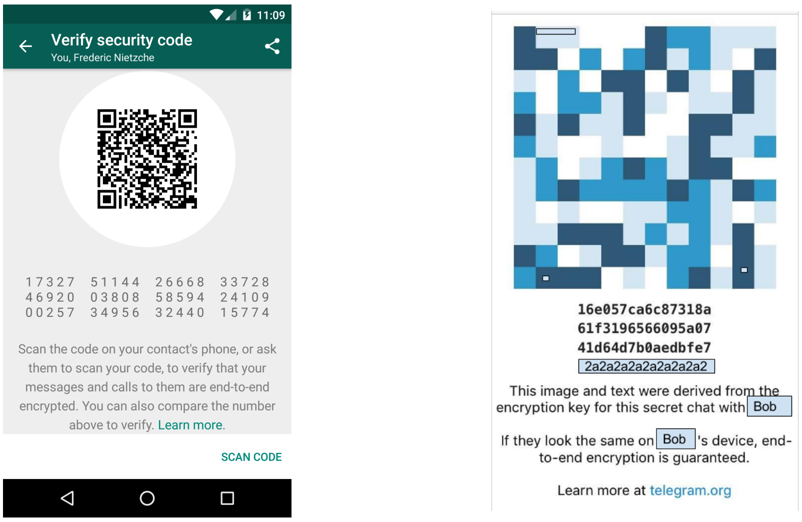
October 13, 2016

Abstract

Some secure messaging applications available on the Internet provide identity verification of participant. For verifying group member's identity users need to compare provided fingerprint or scan QR code. In this report, we present identity verification of Signal secure messaging application that provides the user with participant's fingerprint and gives user two choices for verification of group member's identity. In particular, we introduce how usability of identity verification can be improved. We discuss and integrate SafeSlinger exchange protocol to Signal for exchanging fingerprints and verifying identity of participants.

1 Introduction

There are secure messaging applications available on the Internet that users can use for communication. Some of these tools require users to compare long fingerprints if they want to verify the identity of messaging participant. In case, users physically located in different places they can call to each other and do the comparison by reading fingerprints aloud. Some applications also provide the QR code of group member that can be scanned for identity verification. For example, Figure 1 shows two messaging applications that provide user with identity verification options. Whatsapp is the messaging application where users can verify their participant's identity by scanning provided QR code or comparing key fingerprints as shown in Figure 1a. This makes users be sure that they are really talking to intended person and don't have man-in-the-middle.



```
17327  51144  26668  33728
46920  03808  58594  24109
00257  34956  32440  15774
```

Scan the code on your contact's phone, or ask them to scan your code, to verify that your messages and calls to them are end-to-end encrypted. You can also compare the number above to verify. Learn more.

```
16e057ca6c87318a
61f3196566095a07
41d64d7b0aedbfe7
2a2a2a2a2a2a2a2a2
```

This image and text were derived from the encryption key for this secret chat with Bob

If they look the same on Bob's device, end-to-end encryption is guaranteed.

Learn more at telegram.org

(a) Whatsapp fingerprint verification.

(b) Telegram QR code verification.

Figure 1: Fingerprint and QR code verification examples.

As well as, in Telegram messaging application for verifying identity user needs to compare long fingerprint which was derived from the key that was used to encrypt the conversation with group member as shown in Figure 1b. Here also user can use QR code scanner and scan provided code on participant's phone.

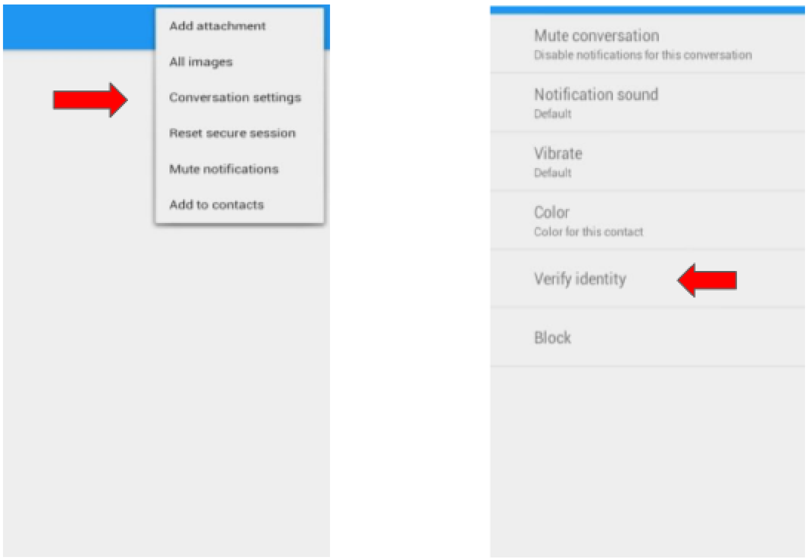
In Signal messaging application user can verify group members' identity by manually comparing fingerprints, as well as, by using QR code scanner to scan provided QR code. However, it is possible to improve usability of verification and use SafeSlinger exchange protocol for exchanging fingerprints of communicating group members, verifying it. After verification user can be informed that participant's identity is correct or not.

In this report, we present identity verification of Signal messaging application and discuss how usability of it can be improved by using SafeSlinger exchange protocol.

2 Signal identity verification

Signal is one of the secure messaging applications that available on the Internet. Communication with some user in Signal is end-to-end-encrypted. Signal server has no access to any user communication and never stores user data. This application provides users with the fingerprint of the participant if they want to verify an identity of the group member, i.e if the user wants to know that really communicates to the correct person. User can verify group member's identity by following the steps below:

1. Select *"Conversation Settings"* from the conversation menu as shown in Figure 2a.
2. In the next page Signal will give user the option *"Verify identity"* that is shown in Figure 2b.
3. As shown in Figure 3 application provides group member's fingerprint for verification.



Mute conversation
Disable notifications for this conversation

Notification sound
Default

Vibrate
Default

Color
Color for this contact

Verify identity

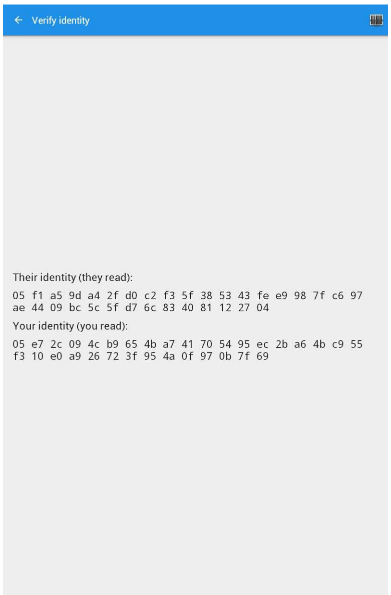
Block

(a) Conversation settings.

(b) Verify identity.

Figure 2: Signal recipient's identity verification.

The user can verify the identity of a participant by comparing each character of fingerprint that is shown in the own device with the fingerprint characters these are shown in participant's device or as shown in Figure 3 in the top right corner user can use QR code scanner for scanning member's code and verify recipient's identity. Once user verified participant's identity Signal will not indicate it in conversation with that member. It is possible that in future user will forget that already has verified this member's identity. This can result in verification of the same participant's identity multiple times.



Their identity (they read):

```
05 f1 a5 9d a4 2f d0 c2 f3 5f 38 53 43 fe e9 98 7f c6 97
ae 44 09 bc 5c 5f d7 6c 83 40 81 12 27 04
```

Your identity (you read):

```
05 e7 2c 09 4c b9 65 4b a7 41 70 54 95 ec 2b a6 4b c9 55
f3 10 e0 a9 26 72 3f 95 4a 0f 97 0b 7f 69
```

Figure 3: Fingerprint view.

However instead of comparing long fingerprints manually and multiple times, Signal can use SafeSlinger exchange protocol for exchanging and verifying identity. As well as, it is possible to indicate already verified users.

3 SafeSlinger library

SafeSlinger library is open source and can be used for exchanging and verifying public keys or other data. It provides secure data exchange. As well as, it protects exchanged data against man-in-the-middle attack during data transformation. Keys are exchanged using a simple server implementation on App Engine.Exchanged data is encrypted using AES in CBC mode with padding. Open source codes available for the SafeSlinger Android client project written in Java, Apple iOS platform client project written in Objective-C, Google App Engine platform server project written in Python. These projects contain: the library that user can add to own Android application; SafeSlinger exchange Developer's application project which shows the minimum requirements to run a SafeSlinger secure exchange; application project source for the SafeSlinger Messenger application.

OpenKeychain is an OpenPGP implementation for Android that used SafeSlinger exchange library for exchanging public keys. It stores your key and key of the people you communicate in your Android. Users needed to perform manual fingerprint comparison for being sure that keys belong to the correct person, keyserver gave the correct key. By exchanging keys with SafeSlinger users can confirm it without performing manual comparisons.

In Signal messaging application users should perform manual comparison of provided fingerprints or scan provided QR code if they want to verify the identity of the persons they are talking to. However, instead of performing manual comparison or QR code scan, we can use SafeSlinger exchange library. With SafeSlinger exchange protocol users' fingerprint can be exchanged securely and can be verified. User could had the options *"Verify identity manually"* and *"Verify identity with SafeSlinger"* in the conversation settings in conversation menu. While using SafeSlinger key exchange protocol instead of verifying long fingerprints manually, the user should perform the comparison of three-word phrases and select the common one. After selecting the same actual word phrase user will be informed about verification result.

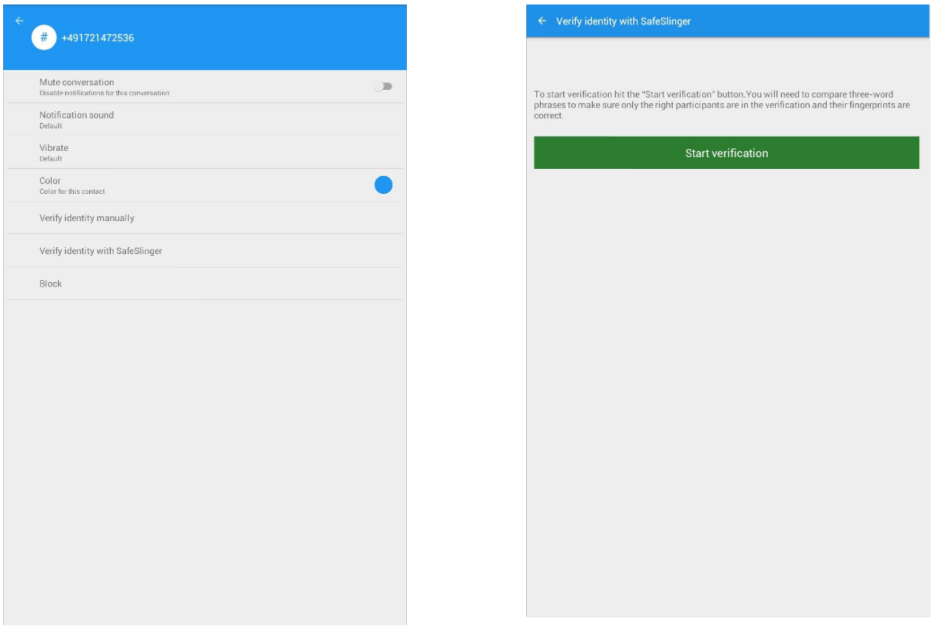
Advantages of using SafeSlinger for identity verification:

- More usable than manual fingerprint verification.
- Secure user data exchange.
- Prevents MitM, GitM attacks.
- Provides user privacy.
- Efficient to use.

4 Integrate SafeSlinger exchange to Signal

Before extending SafeSlinger exchange protocol in Signal we have configured our own Textsecure Server, Push Server and database. As well as, for having trusted communication between servers and connected devices we created our own certificates these were copied to Signal and added to the list of trusted certificates in a device. In Signal, we modified conversation settings menu, where the user had *"Verify identity"* choice for identity verification. Instead of that we added *"Verify identity manually"* and *"Verify identity with SafeSlinger"* these are shown in Figure 4a.

If user selects manual verification from the list then will need to compare fingerprints manually. If the user selects verification with SafeSlinger



+491721472536

Mute conversation
Disable notifications for this conversation

Notification sound
Default

Vibrate
Default

Color
Color for this contact

To start verification hit the "Start verification" button.You will need to compare three-word phrases to make sure only the right participants are in the verification and their fingerprints are correct.

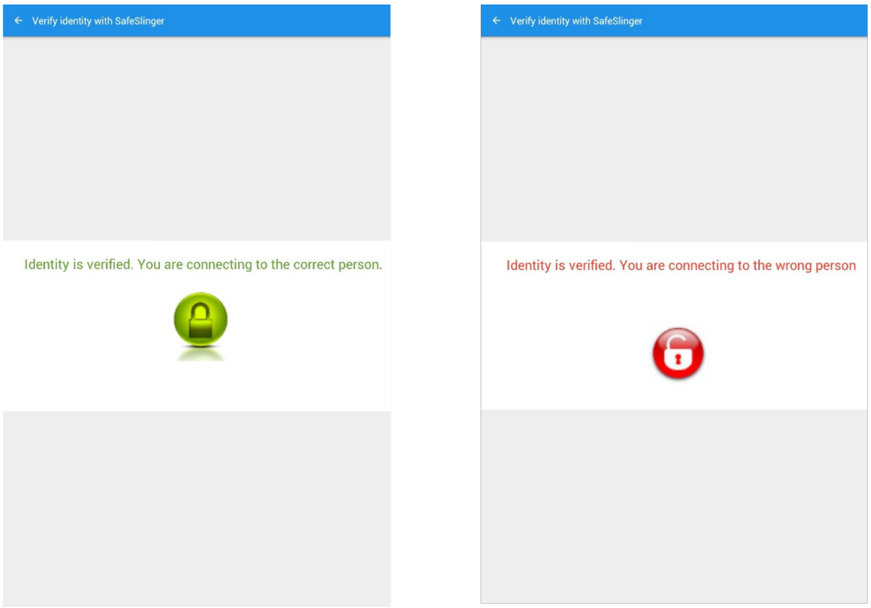
#
Verify identity manually
Verify identity with SafeSlinger
Block
Verify identity with SafeSlinger
Start verification

(a) Conversation Settings.

(b) Start Verification.

Figure 4: SafeSlinger integration to Signal.

then SafeSlinger protocol will be used for the exchange of fingerprint and verification of identity. After selecting *"Verify identity with SafeSlinger"* user will be informed that for starting identity verification should click on *"Start Verification"* button that is shown in Figure 4b and during verification procedure user will need to compare three-word phrases and participants in a verification should select the same common word phrase. When the user clicks on "Start Verification" button SafeSlinger key exchange protocol starts. Users' fingerprints are exchanged. During exchange user's fingerprint is encrypted. In SafeSlinger, users initially select the data that they want to exchange and enter number of group participants in the exchange. Then device generates two random integer values, encrypts shared data and finally from these data calculates commitment and then sends it to the server. The server initially does not know which devices belong to the same group and assigns unique integer ID to each group member. Participants compare those integers and send to the server the minimum one. After that server knows which participants are in the same group and distributes commitment and IDs. Devices compute three-word phrases where only one phrase is same in all devices. An overview video of how SafeSlinger works is available at: http://www.youtube.com/watch?v=IFXL8fUqNKY. Unlike SafeSlinger, in Signal users are already in the same group and know each other before starting the exchange. That is why there is no need to ask the user to enter a number of group participants, as well as, to compare random integers and enter the minimum number for informing SafeSlinger server that these group members are in the same group. We send fingerprint, number of group members in exchange, "Group Name" and "Attempt Name" values to the server. "Group Name" should be unique for each group and each group member must know this value. We used group members' fingerprints to construct this value and then sorted it. "Attempt Name" is an integer number that should be the same for all group members when exchange starts. The "Group Name" and "Attempt Name" are joined and then hashed with Secure-Hash-Algorithm(SHA) 3. Using first 32 bit of hash group ID is created. This ID is unique and globally avoids a collision. So, the user experience will ask for the three-word phrases comparison only.


← Verify identity with SafeSlinger
← Verify identity with SafeSlinger

Identity is verified. You are connecting to the correct person.

Identity is verified. You are connecting to the wrong person

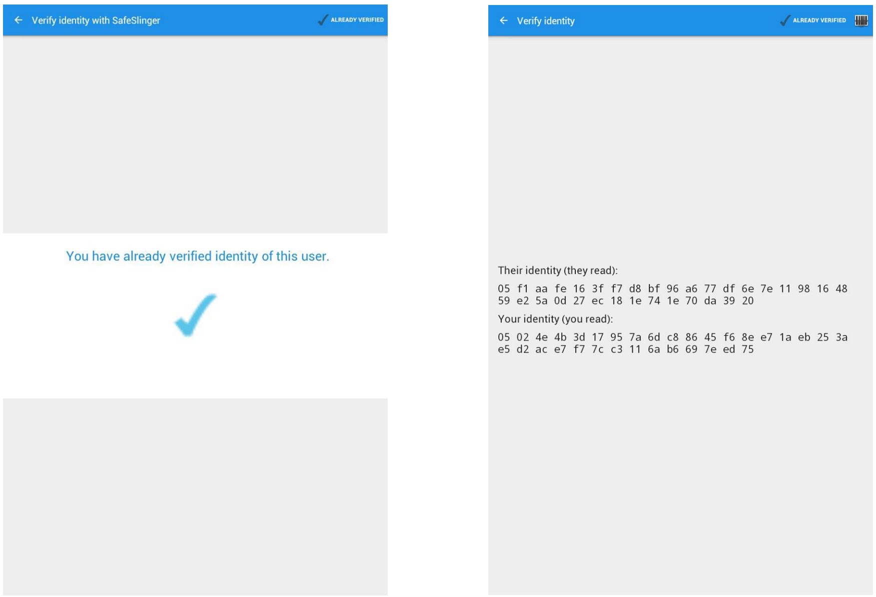
(a) Correct identity.

(b) Incorrect identity.

Figure 5: Signal identity verification.

During exchange each device computes the hash. This hash is represented to the user as a three-word phrase which was selected from PGP word list. This word phrase is actual word phrase. Two additional decoy word phrases also generated. Users should compare represented three-word phrases and select the common one. If all group members in the exchange select the correct actual word phrase then devices in a group engage in group Diffie-Hellman protocol and derive group secret key. This key is used to distribute decryption key between participants. After getting decryption key each device decrypts shared data, these are fingerprints of participants.

That data is stored and compared with the fingerprint that participant already had for a member in verification procedure. If the received fingerprint is the same as the fingerprint user has on the device then identity is correct that is shown in Figure 5a . The user informed that group participant's identity is verified and connects to the correct person. As well as, this group member's identity is indicated as verified by having "Already verified" check in the SafeSlinger verification page. As well as, this check added to "Verify identity manually" page. If the received fingerprint is not the same as the possessed fingerprint then the user will be warned that fingerprints are not the same and communication with this member is not secure that is shown in Figure 5b.



You have already verified identity of this user.

Their identity (they read):

```
05 f1 aa fe 16 3f f7 d8 bf 96 a6 77 df 6e 7e 11 98 16 48
59 e2 5a 0d 27 ec 18 1e 74 1e 70 da 39 20
```

Your identity (you read):

```
05 02 4e 4b 3d 17 95 7a 6d c8 86 45 f6 8e e7 1a eb 25 3a
e5 d2 ac e7 f7 7c c3 11 6a b6 69 7e ed 75
```

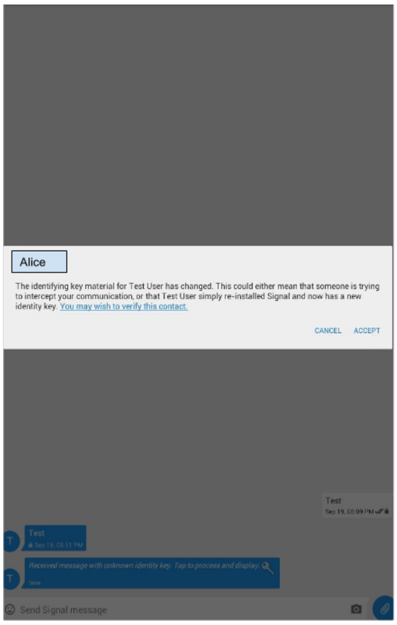
(a) Already verified user.

(b) Verified user.

Figure 6: Signal verified user.

After verifying group member's identity and getting informed that identity is correct, the user doesn't need to verify this group member's identity again. We indicate already verified group member in the top right corner of the "Verify identity with SafeSlinger" and "Verify identity manually" pages. In case group member's identity is already verified and the user clicks on "Verify identity with SafeSlinger" then Figure 6a will be shown that you have already verified this user's identity and identity is correct. The user can click on "Verify identity manually" and in right top corner will see that this user's identity is already verified as shown in Figure 6b. But still can access QR code verification and can see the fingerprint of a group member. If user verifies group member's identity by scanning provided QR code of group member, then "Already verified" check will be added to the same pages.

In case, participant's fingerprint is changed Signal will inform the user that group member's key has changed as shown in Figure 7.



The identifying key material for Test User has changed. This could either mean that someone is trying to intercept your communication, or that Test User simply re-installed Signal and now has a new identity key. You may wish to verify this contact.

Test
Sep 19, 05:09 PM

Test
Sep 19, 05:11 PM

Received message with unknown identity key. Tap to process and display.
Now

Figure 7: User's fingerprint has changed.

In this case, "Already verified" check in verification pages will be removed and the user will need to verify that member's identity again. The user can accept new identity key or can decline it. In case, user declines new identity key, will not receive messages from that group member. If the user accepts new identity key then can verify new identity with provided methods, i.e SafeSlinger, manual comparison of fingerprints and QR code scan.

5 Conclusion & Future work

In this report, we presented secure Signal messaging application's identity verification methods. In particular, we integrated SafeSlinger exchange protocol to Signal for fingerprint exchange and identity verification. Verifying identity using SafeSlinger exchange and verification is the most usable method between available fingerprint-based identity verification methods. SafeSlinger exchanges fingerprints of users' securely and privately. It asks members to compare three-word phrases and choose the phrase that is same on all devices. It prevents MitM, GitM attacks during fingerprint exchange. Only group members can get exchanged data.

In Signal messaging system, users should perform long fingerprint comparison if they want to verify the identity of a person they are talking to. However, the usability of identity verification is improved by using SafeSlinger exchange protocol. User's fingerprint is exchanged and verified with SafeSlinger. A user informed about verification result. We have used "Already verified" check in verification pages to indicate secure conversation with group member and to indicate that this group member's identity has already verified. The user won't need to verify this member's identity again. But if group member's identity is changed, user will be informed about it and "Already verified" check those were in the verification pages will be removed. In this case, the user will need to verify this group member's identity again.

SafeSlinger exchange protocol also supports an exchange of up to ten users' data at the same time. In Signal, we implemented fingerprint exchange of two users. But exchanging fingerprints of multiple users at the same time can be done in the future using SafeSlinger's that feature.

References

[1] SafeSlinger Project. `https://github.com/SafeSlingerProject`.

[2] WhisperSystems/signal-android: A private messenger for android. `https://github.com/WhisperSystems/Signal-Android`.

[3] Y.-H. Lin T. H.-J. Kim J. McCune , M. Farb and A. Perrig. 'SafeSlinger: Easy-to-use and secure public-key exchange,'. *in International Conference on Mobile Computing Networking*, pages 417–428, ACM, 2013.