Trusted Introductions For Secure Messaging*

Christelle Gloor[0000−0001−7031−2577] and Adrian Perrig[0000−0002−5280−5412]

Network Security Group, D-INFK, ETH Zürich, Switzerland
Christelle.gloor@inf.ethz.ch, adrian.perrig@inf.ethz.ch
https://netsec.ethz.ch/

Abstract. Today's prevalent end-to-end encrypted messaging platforms using the Signal Protocol, which brought opportunistic encryption and resistance to eavesdropping, **are still vulnerable to impersonation attacks**. We propose **Trusted Introductions**, a mechanism to *transfer existing trust ties between users*, built on the user verification capabilities of the Signal Protocol, to increase resistance to active attacks. We argue that replacing user-managed identity-keys in cryptography with the introduction concept, will increase users' understanding and *improve usability* of the verification mechanism. Current events underscore the need for *anonymous introductions*, which can be achieved based on the Signal Protocol's properties of forward secrecy and deniability.

Keywords: Usability · Public Key Cryptography · Identity Binding · Trust Transfer · Private Messaging · the Signal Protocol · Safety Number Verification

1 Problem Statement

Alongside smartphones, private messaging became universally available. End-to-end encryption in private messaging systems emerged from concerns about privacy and a lack of trust in application providers and operators. The Double Ratchet Algorithm developed in 2013 and used by the Signal Protocol provides forward secrecy [11, 12], and is widely adopted. Major private messaging applications, originally envisioned as a free and/or secure replacement to heavily surveilled SMS, presently rely on the protocol, collectively serving billions of users [2, 9, 13, 18, 21].

The Signal Protocol minimizes necessary trust in the centralized operational messaging infrastructure, by decreasing the amount of data the infrastructure retains about its users [8]. Consequently, when trying to connect to another user after having fetched cryptographic material from the infrastructure the protocol provides privacy but no guarantee w.r.t whom one is communicating with [14]. **Impersonation and other active attacks are therefore fundamental vulnerabilities.**

* We gratefully acknowledge support for this project from the Werner Siemens Stiftung (WSS) Centre for Cyber Trust at ETH Zurich. https://cyber-trust.org

Users must verify identity to ensure the absence of impersonation attacks. Most commonly, users perform the verification through bilateral QR-code scans, or manual comparisons of safety numbers. The safety number is a concatenation of hashes of both participants' public identity keys and unique identifiers, thus distinct for each pair of users, and must be equal on both clients for verification to succeed [6].

Anecdotally, not many users perform this additional step, as verifying each contact is cumbersome and most users are unaware of the benefits.

We therefore propose a mechanism to *transfer previously established trust to another user,* thus improving usability, maximizing the benefit of each verification and increasing resistance of the messaging system against impersonation attacks.

We present the trust transfer mechanism from the perspective of safety number verification. We then consider the following questions: (1) In the presence of an oppressive regime performing active attacks, which security guarantees are essential? (2) What level of security can be achieved with the proposed mechanism? (3) How does the mechanism compare to alternative proposals?

2 A Trusted Introduction

Alice and Bob have met up to verify their safety number. Bob would like to securely get in touch with Carol, but is concerned about impersonators, while being unable to personally meet up with Carol. Bob is aware that Alice knows Carol. Bob asks Alice, who has previously verified Carol, for a *trusted introduction.*

2.1 Background

The safety number computation varies between applications, but minimally contains a hash of both users' public keys and unique identifiers. For simplicity, we will consider the calculation performed by the Signal client [6]. When a user registers with the server and control of the provided phone number (used for DoS protection and contact discovery) is verified, the client creates a key-pair and a unique identifier is assigned by the server [5]. Subsequently, the phone number may be changed as it is not part of a user's identity. The human readable safety number is a numerically ordered concatenation of the identity digest of both parties [16]. Each digest is a repeated SHA-512 hash over the version, unique identifier and public key of the party, truncated to a 30-digit decimal number, comprising half of the safety number. The 60-digit safety number is identical for both parties. Faking the safety number would involve finding a hash collision for both digests. This is computationally intractable based on the collision resistance properties of the hash function. Thus, identical safety numbers on both clients confirm the absence of a third party and associated attacks.

The safety number between two parties is computable, if and only if the public keys and identities are known. Alice verified them for both Bob and Carol.

2.2 Proposal

To perform the **trusted introduction**, Alice forwards her computation of the safety number between Carol and Bob to Bob over their verified secure channel. If Bob's client's computation of the safety number matches what is sent by Alice, and he trusts Alice, Bob is assured to be communicating with the account that Alice has verified as Carol's. If the number does not match, he got served a malicious public key and/or unique identifier by a compromised infrastructure and can detect the attack.

Bob evaluates the trustworthiness of an introduction solely by weighing the trust he has in the introducer (Alice), which is an intuitive mapping to human relations and networking in the offline world. Analogously, we keep the requests and initiations of introductions purely within human relationships, instead of automating.

Additionally, we believe it to be beneficial to impose the limitation that only safety numbers directly verified by the user may be forwarded through a *trusted introduction*, and not safety numbers that have been introduced to the user. "Introduction chains" for which some hops are unknown to the recipient, are non-trivial to assess and may leak a partial social graph of participants on the chain. This restriction achieves the property of *limiting the damage of a malicious introduction to the direct contacts of the malicious introducer*. In contrast, this limits the spread of valid introductions that may be difficult to find a direct contact for. Further research will be needed to evaluated the risk/benefit relation of both approaches, but we initially choose to err on the side of caution and simplicity.

If there is a requirement of anonymity, (Bob may have a reason to obscure the information that Alice introduced him to Carol, anticipating a future breach of his phone), the introducer information may be purged without trace, leaving an *anonymous* introduction. This is enabled by the forward secrecy and deniability properties of the Signal Protocol [14].

3 Why The Signal Protocol?

The Signal Protocol and its applications are highly relevant since billions of people, through apps like WhatsApp, Signal and Facebook Messenger, rely on it daily. Given this large user base, the paradigm of opportunistic encryption with no initial overhead to the user has proven practical. While the protocol appears to "just work" from a user's perspective, the lack of authentication can have far reaching consequences.

For example, one could imagine the covert large-scale insertion of surveillance backdoors into our most intimate communication systems [7]. Eliminating or reducing this risk would increase the privacy of a large percentage of private communications.

Additionally, *the infrastructure has matured* and issues that were traditionally difficult to solve, such as key revocations, are resolved in practice. What is

shown to the user when a revocation occurs varies between applications, but both WhatsApp and Signal show a banner in the conversation warning that the safety number has changed. The majority of users are unaware of the implications and will likely ignore the warning due to its frequent occurrence.

However, **the revocation logic will expire all introductions made for the user whose key has been revoked**. Saving a record of expired introductions allows users to request a fresh introduction from the previous introducer – a concrete action to conveniently re-establish trust. **The introduction mechanism does not require any changes to the underlying cryptographic protocol**. Therefore, the typical messaging experience stays untouched, preserving usability while offering an additional layer of security for users with increased privacy needs.

4 Properties in Context

Let's consider the Iranian protests of 2022 to contextualize the discussion and examine a threat actor that may **(1) infiltrate the central operational infrastructure to stage an active attack, (2) attempt to covertly infiltrate sensitive conversations, and (3) breach protesters mobile phones after delicate conversations took place to prosecute co-conspirators**.

In this high stakes situation where the government is brutally suppressing efforts of the people to organize, *resistance to passive eavesdropping* is of paramount importance, a property already achieved by the Signal Protocol. This is, however, insufficient, since the government may still stage *active attacks*. Being identifiable with a real identity, e.g., through a phone number registered to one's name, can be lethal [1]. But the need to communicate persists, making anonymous handles (e.g., by using a prepaid SIM anywhere in the world) a viable option.

Even if people are anonymous, infiltrated conversations may lead to a disruption of plans. Thus, *verification preserving anonymity* prior to sensitive conversations is desirable. *A trusted introduction must not tie to a real identity.* Instead, we built relative trust, only anchored to the possession of the private key. The validity of the introduction can be reasoned about in the same way a person would reason about an offline introduction, hinging on the trustworthiness of the introducer.

The user retains full control about the information they are willing to share– nothing is exchanged in obscurity. If the introducer information is sensitive and anonymity is more important than convenient re-establishment of trust, this information can be deleted, leaving behind an *anonymous introduction.*

5 Related Work

Pretty Good Privacy (PGP), was developed by Philip Zimmermann in 1991 to provide public key based private and authenticated email and make encryption widely available [23]. While PGP still exists, it is widely understood to have missed the original vision. It spawned numerous papers analyzing its usability

[17, 19, 22] and opinion pieces from avid proponents of encryption on why PGP needs to go [4, 10, 15, 20].

While the trusted introductions are related to PGP and the Web of Trust, there are some key differences: PGP attempted to build a global web of endorsements tied to real identities. This incentivized people to sign and endorse keys indiscriminately and spread this information as widely as possible, e.g., by posting the information to key directories, to increase the connectivity of the web. The Trusted Introductions mechanism works on a more local scale and introductions are by nature ephemeral and relative. Finally, there was no notion of anonymous introductions or a practical revocation system in PGP.

Safeslinger was one of the first mobile applications enabling key exchanges for end-to-end encryption [3]. The focus of the paper is on efficient and secure group key exchange, and the application subsequently allowed to use the keys for encrypted messaging and file transfer. The application is developer centric, expecting the user to understand and manage cryptographic keys. Trusted introductions (called *secure* introductions) were proposed as a bidirectional operation where an introduction forwards the information to both contacts verified by the introducer. There was no notion of anonymous introductions. Revocations, while mentioned, remain unsolved.

6 Discussion

We have progressed from the first public key crypto-systems, that were widely regarded as unusable, to the user-perceived "just-works" paradigm we have today. Work still needs to be done in threat communication and educating users about the limitation of opportunistic encryption and how to overcome them.

Various trade-offs arise, such as the previously mentioned ability to forward introductions and therefore spread secure information, versus the proposed limited introduction mechanism of direct verifications, which allows for sound reasoning on the receiving end.

Gamification of introductions could promote the mechanism. However, this may also result in forwarding untrusted introductions to "win the game", therefore diluting the benefits and possibly undermining trust in the system.

We must additionally take care to avoid implying to users that the introductions mechanism comes with fundamental changes, thus suggesting that it is no longer safe to use the messaging system without introductions. Failure to do so may lead to an exodus to less secure systems which feign security by being less transparent about their weaknesses.

Further research must compare relative bindings as achieved in this proposal with the absolute bindings that we are attempting today. How do the achieved properties differ and which problems can we solve through this approach?

User trials and feedback will show what works, which will ultimately be the decisive factor on adoption and success of this proposal.

7 Future Work

We are in the process of finalizing a client-side prototype implementation in the open source Signal messenger, aiming to provide the basis for further research.

Acknowledgements We gratefully acknowledge support for this project from the Werner Siemens Stiftung (WSS) Centre for Cyber Trust at ETH Zurich https://cyber-trust.org, and thank Giacomo Giuliari for his valued feedback.

References

1. Iran: Death Sentences Against Protesters (Dec 2022), https://www.hrw.org/news/2022/12/13/iran-death-sentences-against-protesters
2. Facebook: Messenger secret conversations, https://about.fb.com/wp-content/uploads/2016/07/messenger-secret-conversations-technical-whitepaper.pdf
3. Farb, M., Lin, Y.H., Kim, T.H.J., McCune, J., Perrig, A.: SafeSlinger: easy-to-use and secure public-key exchange. In: Proceedings of the 19th annual international conference on Mobile computing & networking - MobiCom '13. p. 417. ACM Press, Miami, Florida, USA (2013). https://doi.org/10.1145/2500423.2500428, http://dl.acm.org/citation.cfm?doid=2500423.2500428
4. Green, M.: What's the matter with pgp?, https://blog.cryptographyengineering.com/2014/08/13/whats-matter-with-pgp/
5. Greyson Parrelli, J.R.: Android implementation signal service id, https://github.com/signalapp/Signal-Android/blob/cb0e7ade141fc9b1c707d53c52cc2ab5b784207b/libsignal/service/src/main/java/org/whispersystems/signalservice/api/push/ServiceId.java
6. Greyson Parrelli, Jordan Rose, n.b.C.H.A.H.: Android implementation security numbers, https://github.com/signalapp/Signal-Android/blob/main/app/src/main/java/org/thoughtcrime/securesms/verify/VerifyDisplayFragment.java
7. Ian Levy, C.R.: Principles for a more informed exceptional access debate, https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate
8. jlund: Technology preview: Sealed sender for signal, https://signal.org/blog/sealed-sender/
9. Marlinspike, M.: Facebook messenger deploys signal protocol for end-to-end encryption, https://signal.org/blog/facebook-messenger/
10. Marlinspike, M.: Gpg and me, https://moxie.org/2015/02/24/gpg-and-me.html
11. Marlinspike, M.: Signal on the outside, signal on the inside, https://signal.org/blog/signal-inside-and-out/
12. Marlinspike, M.: Textsecure, now with 10 million more users, https://signal.org/blog/cyanogen-integration/
13. Marlinspike, M.: Whatsapp's signal protocol integration is now complete, https://signal.org/blog/whatsapp-complete/
14. Marlinspike, M.: The x3dh key agreement protocol, https://signal.org/docs/specifications/x3dh/#identity-binding
15. Perry, M.: [tor-talk] why the web of trust sucks, https://lists.torproject.org/pipermail/tor-talk/2013-September/030235.html

16. Rose, J.: Rust implementation fingerprint generation, https://github.com/signalapp/libsignal/blob/main/rust/protocol/src/fingerprint.rs#L154
17. 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 - SOUPS '13. p. 1. ACM Press, Newcastle, United Kingdom (2013). https://doi.org/10.1145/2501604.2501609, http://dl.acm.org/citation.cfm?doid=2501604.2501609
18. Signal: Signal technical information, https://signal.org/docs/
19. Tong, W., Gold, S., Gichohi, S., Roman, M., Frankle, J.: Why King George III Can Encrypt p. 13
20. Valsorda, F.: Op-ed: I'm throwing in the towel on pgp, and i work in security, https://arstechnica.com/information-technology/2016/12/op-ed-im-giving-up-on-pgp
21. WhatsApp: Whatsapp encryption overview, https://faq.whatsapp.com/820124435853543
22. Whitten, A., Tygar, J.D.: Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In: Proceedings of the 8th conference on USENIX Security Symposium - Volume 8. p. 14. SSYM'99, USENIX Association, USA (Aug 1999)
23. Zimmermann, P.: Why i wrote pgp, https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html