Trusted Introductions For Secure Messaging[*]

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

Department of Computer Science, Network Security Group, ETH Zürich
christelle.gloor@inf.ethz.ch, adrian.perrig@inf.ethz.ch
https://netsec.ethz.ch/

Abstract. Although today's most prevalent end-to-end encrypted messaging platforms using the Signal Protocol perform opportunistic encryption and provide resistance to eavesdropping, **they are still vulnerable to impersonation attacks**. We propose **Trusted Introductions**, a mechanism to *transfer existing identity verifications between users*, to increase resistance to active attacks. The proposal builds on the out-of-band user identity verification capabilities provided by the Signal Protocol. We argue that replacing user-managed identity-keys in cryptographic systems with the concept of an introduction, 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 repudiation.

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

1 Problem Statement

With the broad adoption of smartphones, encrypted messaging became universally available. End-to-end encryption in encrypted messaging systems emerged from concerns about privacy and a lack of trust in network and infrastructure providers and operators. The Double Ratchet Algorithm developed in 2013 and used by the Signal Protocol provides forward secrecy [12, 13], and is widely adopted. Major secure messaging applications, originally envisioned as a free and/or private replacement to heavily surveilled SMS, presently rely on the protocol, collectively serving billions of users [3, 10, 14, 20, 23].

The Signal Protocol minimizes necessary trust in the centralized operational messaging infrastructure, by decreasing the amount of data the infrastructure retains about its users [9]. Consequently, when trying to connect to another user after having fetched cryptographic material from the centralized server, the protocol provides privacy, but no guarantee regarding with whom one is communicating [15]. There is no in-band mechanism hindering a compromised

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

server from dishonestly answering a request for cryptographic material, thus connecting the user to an adversary instead of the expected communication partner. **Impersonation and other active attacks are fundamental vulnerabilities.**

Users must verify the identity of their communication partners 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 [16].

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 identity verifications to another user,* thus improving usability, maximizing the benefit of each verification, and increasing resistance of the messaging system against impersonation attacks.

We first consider which security guarantees are essential in the presence of an oppressive regime performing active attacks. Next, we present our trust transfer mechanism from the perspective of safety number verification and analyse which security guarantees can be achieved with the proposed mechanism. Finally, we compare our mechanism with alternative proposals.

2 Use Case

Let's consider the Iranian protests of 2022 to put the discussion in context 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 have taken place**. The adversary does not break cryptographic primitives, nor do we consider the leak of private keys or more general breach of mobile phones while the device is actively used for sensitive communications.

In this high stakes situation where the government is suppressing efforts of people to organize and attempting to persecute conspirators, *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 pseudonymous handles (e.g., by using a prepaid SIM anywhere in the world) a viable option.

Even if people are unidentified, infiltrated conversations may lead to a disruption of their plans to protest. We must ensure that our proposal does not necessitate a tie to a *recognized identity*, for example, government issued IDs. We do not want to build globally valid endorsements. Instead, we built *relative trust*, only anchored to the possession of the private key verified by contacts we know and trust. The user must have the ability to reason about the validity of the trust transfer. We achieve this in the same way a person would reason about an offline introduction: by reasoning about the trustworthiness of the introducer.

Finally, *the user must retain full control about the information they are willing to share*–no information should be exchanged in obscurity, or sensitive information may land in the adversary's hands. If the introducer's identity is sensitive and their anonymity is more important than convenient re-establishment of trust, this information can be deleted, leaving behind an *anonymous introduction*.

3 A Trusted Introduction

Alice and Bob both use the Signal Protocol and met in person 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 with Carol. Bob is aware that Alice knows Carol. Bob asks Alice, *who has previously verified Carol*, for a *trusted introduction*.

3.1 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 [16]. When a user first registers, the server will verify that the user controls the provided phone number. This is done for denial of service (DoS) protection and contact discovery and does not entail authentication. The client then creates a key pair and sends the public key to the server. The server generates a unique identifier for the client and stores the association with the public key and phone number [7]. 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 [18]. Each digest is a repeated SHA-512 hash over the version, unique identifier, and public key of the party. The digest is then truncated to a 30-digit decimal number. This comprises half of the safety number, and an ascending ordering of the two halves results in an identical 60-digit safety number for both parties. Faking the safety number involves 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 in their communication channel and associated attacks.

The safety number between two parties is computable, if and only if the public keys and identities are known. *Alice must have securely obtained them for both Bob and Carol before making an introduction.*

3.2 Proposal

To perform the **trusted introduction**, Alice forwards Carols contact details and the computation of the safety number between Carol and Bob to Bob over their verified channel. Note that, in this as well as most cases, the introduction may be bilateral, but it is not mandated. If Bob's client's computation of the safety number with Carol matches the value that 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 has been served a compromised public key and/or unique identifier by an impostor (which compromised Bob or the infrastructure) and can detect the attack.

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

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 for which it may be difficult to find a direct contact. Further research will be required to evaluate the risk/benefit trade-off of both approaches, but we chose to initially favour a simple and cautious approach.

If the introducer desires anonymity from parties other than the introduction recipient, the introducer information may be purged without trace, leaving an *anonymous* introduction. This is enabled by the forward secrecy and repudiation properties of the Signal Protocol [15].

We provide a more detailed reasoning for the design decisions and infer the security guarantees of the proposal in the remainder of this chapter.

3.3 Protocol Properties

Let there be three protocol participants, Bob, Alice, and Carol who wish to verify each other. We denote the half of the safety number associated with a participant by the first letter of their name. Recall that both the public key and the user's identity is encoded in their half of the safety number.

Goal:

The protocol allows a participant to forward the verification of a second party to a third party through an *introduction.* For our example, Alice wants to forward her verification of Carol to Bob.

We denote the connection that Alice has with Carol as the *verification path* and the connection between Alice and Bob as the *forwarding path* of the introduction.

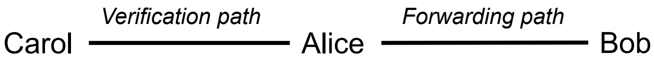


Fig. 1: The two connections relevant to an introduction. The information flows from Carol to Bob, via Alice.

Different actions result in different levels of confidence for each path, which determine the further actions that are permitted by the implementing client. We model three confidence levels. A participant can directly verify (D) the safety number of a second party by scanning their QR code, a participant may be verified through an introduction (I) over a secure channel, or there may be no verification (N). For the purpose of this paper, we assume the direct verification to be carried out by a QR-code scan in person and thus be secure, even though in practice this is unenforced. Since an introduction involves faith in the introducer who may be colluding with an adversary, a total ordering is defined as follows:

$$DirectlyVerified > Introduced > NoVerification$$

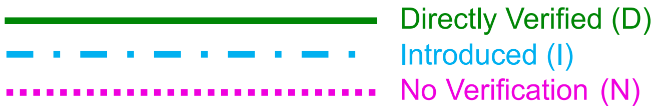


Fig. 2: A connection between two participants (a possible path of an introduction) may have one of three different confidence levels that are represented pictorially throughout the remainder of this paper.

We further assume that a confidence relation is bidirectional. This may not be the case in practice, since Alice can prevent Carol from knowing that she introduced her to Bob by making a unidirectional introduction. But an honest and correct introduction implies the absence of a third party on the introduced path. This global property is reflexive and independent of missing user knowledge resulting from unidirectional introductions. A more detailed analysis, taking varying user perspectives and further variables into account, is left for future work.

Preconditions:

- The introducing party, Alice, MUST have a strong confidence relation ($ConfidenceLevel == D$) with the introducee, Carol.
- There MUST exist a secure channel ($ConfidenceLevel > N$) between the introducer, Alice, and the receiving party, Bob as shown in figures 3 and 4.

Note that these preconditions may prevent a bilateral introduction if the forwarding path and verification path have unequal confidence levels.

Mechanism:

1. Bob asks Alice to introduce Carol to him.
2. Alice computes the expected safety number between Bob and Carol, BC.
3. Alice introduces Carol to Bob by forwarding BC and Carol's contact information to Bob.
4. Bob checks the forwarded safety number against the value served by the infrastructure. If the values conflict and Bob trusts Alice, he has discovered the presence of an active attack on the path between him and Carol.

Note that a lying introducer could attempt a DoS attack and sow general mistrust by distributing incorrect introductions. Comparing multiple introductions for the same introducee by utilizing a Byzantine fault tolerant algorithm may be a helpful mitigation, but we consider this out of scope for this paper. Instead, the introduction recipient bases their decision solely on the trustworthiness of the introducer, and may request additional introductions from alternative introducers if they suspect malice.

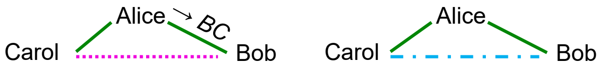


Fig. 3: A legitimate introduction over a forwarding path of confidence D.

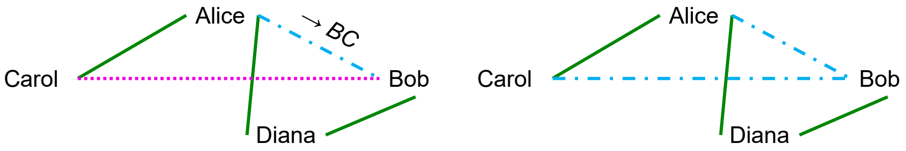


Fig. 4: A legitimate introduction over a forwarding path of confidence I.

Anonymous Introductions: If the identity of an introducer is sensitive information, this may be scrubbed by the client after its evaluation, leaving behind an anonymous introduction.

This is trivially executed on the introduction recipients' client, where the introduction is stored. Nevertheless, *fundamentally there cannot be any cryptographic enforcement of anonymity if the identity of the introducer is used to assess the introduction.*

Alice, who is likely the one most harmed by the leakage of her identity, must trust Bob's word that this information will be deleted. While Bob may always choose to retain this information without Alice's consent, the case of non-malicious carelessness by Bob can be alleviated by automation.

Alice may send an introduction with the condition that she not be retained as the introducer. Since the recipient will accept or reject introductions based on the trust they have in the introducer, the information should not disappear before Bob has the chance to make a decision. However, inaction by Bob should not indefinitely delay the deletion of Alice's information.

To solve this, we propose a *timeout* enforced by the client by which the introduction is deleted completely if Bob did not interact with it. If Bob accepted the introduction, only the introducer information is deleted, preserving the new verification state for Carol.

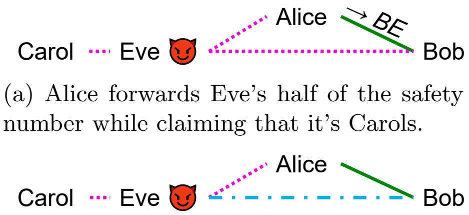
Revocations: The invalidation of introductions employs the Signal Protocol's key revocation mechanism. The centralized infrastructure communicates key changes to users. Once a key changes, any introduction that contained this key for the *introducee*, Carol, will turn stale. This resets the I trust relation between Bob and Carol to N.

Note that a key change of the introducer, Alice, *will not* invalidate introductions that have been made by Alice while her key was still valid. The introduction's forwarding channels serve only as an ephemeral secure means to forward information. Alice *does not* cryptographically sign the introduction, which would warrant invalidation along with her key. Bob decides if, at that moment, he trusts Alice to have done the verification of Carol diligently and to be honest, and then accepts or rejects the introduction. Alice losing her phone later, or being compromised after the introduction already happened, does not change the validity of introductions she made while her key was still valid. An introduction is not an endorsement of recognized identity, instead the introducer is asserting the relative identity of the introducee as the holder of the private key bound to the safety number.

Guarantees:

We further assume that the in-person QR code scan for verification is secure. Under this assumption, an attack may only succeed if introducers actively lie and collude with the adversary Eve, who controls the responses of the server as follows:

Different forwarding paths: We consider two distinct cases. First, if the forwarding path has a **confidence level of** D, an attack will only succeed is if the introducer colludes with the adversary and lies about the safety number of the introducee, as shown in figure 5.

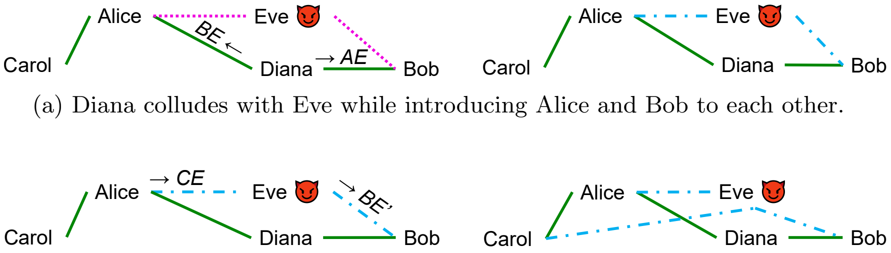


(a) Alice forwards Eve's half of the safety number while claiming that it's Carols.

(b) The created introduction is with Eve instead of Carol and the attack is not detected.

Fig. 5: A successful attack for a forwarding path with confidence level D.

If the forwarding path has a **confidence level of** I, even a honestly forwarded introduction may be changed in transit if the introducer of the forwarding path was colluding with the adversary at the time of the preceding introduction.



(a) Diana colludes with Eve while introducing Alice and Bob to each other.

(b) Even though Alice is honest, Eve can now hide her presence.

Fig. 6: A successful attack for a forwarding path with confidence level I.

Bob must now evaluate Alice's trustworthiness as well as Diana's during the initial introduction of Alice. The risk is increased since the collusion could have occurred in more than one place and the trust assumption depends on two actors.

While this weakened condition is riskier, the trade-off between security and accessibility must be balanced. Restricting introductions to only use D paths, while more secure, significantly restricts the number of introductions an individual may initiate. This limits the spread of useful but expensive direct verifications and may present a significant hurdle to adoption.

Given that Bob has an introduced relationship with Alice, he must have decided to trust Diana at the time. Thus we recommend taking the path of accessibility and allowing forwarding over introduced paths. Additionally, one can strengthen the introduced forwarding paths by collecting multiple non-anonymous introductions for the same path. Each additional introduction adds a user that must have colluded with the adversary for the attack to succeed. If this level of security does not suffice, users may individually decide to disallow introductions forwarded over an introduced path, an option which may be offered as an opt-in setting.

We currently recommend that clients implement the restriction of only allowing the introduction of contacts that have been verified in person, as shown in figure 7. Without this, faked introductions could further propagate through the network. This comes at a similar accessibility cost to limiting forwarding paths. It also implies that the distinction between an introduced contact and a directly verified contact must be kept in the users' client, information that could be sensitive if the phone was breached at a later time.

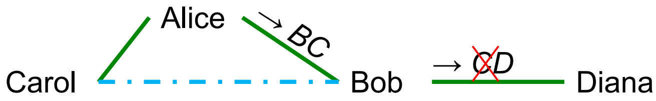


Fig. 7: Diana cannot receive introductions for contacts that have a larger distance than one hop from the introducer.

If people freely forward introductions, a fake introduction can spread through the network without bounds. The adversary only needs to find one contact of the target to collude with, to possibly spread her safety number instead of Carol's, and create a false sense of security.

Additionally, the trustworthiness of an introduction is no longer solely anchored in the introducers Bob has decided to trust. Chaining introductions implicitly requires trust in any person on the introduction path to Carol (or Eve), which may not be known to Bob. Revealing this information would leak a partial social graph weakening privacy.

In contrast, with the proposed client-side restriction in place, a faked introduction may only travel at most one hop, as shown in figure 8. We limit the damage to the direct contacts of the colluders — a desirable property. Note again, that *there is no cryptographic enforcement of this restriction.* We leave room for discussion as further research will be needed for a definitive answer.

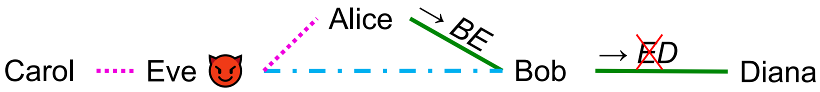


Fig. 8: A faked introduction can only reach the direct contacts of the colluding introducer.

Recall figure 6, and note that allowing the forwarding of introductions over introduced paths does not break this guarantee. Only contacts directly connected with the compromised path are affected, since the incorrect introduction may not be forwarded.

In summary:

1. An attack spread over forwarding paths of confidence level D is guaranteed to be detected if the introduction recipient can find at least one alternative honest introducer for the same target.
2. An attack spread over a compromised forwarding path of confidence level I is guaranteed to be detected if there exists at least one other uncompromised path with an honest introducer for the same target and recipient.
3. A successful attack cannot spread beyond the direct contacts of the introducer colluding with the adversary.
4. The trustworthiness of an introduction can be considered in relation to the trustworthiness of introducers known to the introduction recipient.

4 Related Work

Pretty Good Privacy (PGP) was developed by Philip Zimmermann in 1991 to provide public key based authenticated email and make encryption widely available [25]. While PGP still exists, it is widely understood to have missed the original vision. It spawned numerous papers analyzing its usability [19,21,24] and opinion pieces from avid proponents of encryption on why PGP is unsatisfactory [6,11,17,22].

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, possibly pseudonymous, recognized identities. This incentivized people to sign and endorse keys indiscriminately, spreading them as widely as possible. Endorsements would be invalidated if the key of the endorser were revoked, even though there was no reliable method to propagate revocations. Additionally, users were expected to understand and have the skill to manage long-term cryptographic keys, a highly complicated feat. Finally, endorsements were explicitly kept as evidence supporting the validity of the key. Anonymous endorsements cannot meaningfully exist in this context.

The trusted introductions work on a more local scale. Introductions are ephemeral, relative to introducers, and mirror an everyday concept understood by users.

Safeslinger was one of the first mobile applications enabling key exchanges for end-to-end encryption [4]. The focus of the paper is on efficient and secure group key exchange, and the application subsequently enabled the use of 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, were not available.

5 Discussion

The Signal Protocol and its applications are highly relevant to billions of people through popular apps such as WhatsApp, Signal, and Facebook Messenger. Given this large user base, the paradigm of opportunistic encryption with no initial overhead to the user has proven valuable. While the protocol appears to "just work", the lack of authentication can have far reaching consequences. Work still needs to be done in communicating potential threats and educating users about the limitations of opportunistic encryption.

The covert and overt large-scale insertions of surveillance backdoors into our most trusted communication systems have already been proposed [2,8]. Eliminating or reducing this risk would increase the confidentiality of a large percentage of private communications.

Additionally, *the infrastructure has matured* and problems 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. Still, the majority of users are unaware of the implications limiting the utility of the warning.

However, we can use the revocation logic to **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. **This 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.

With the rise of generative AI systems capable of producing convincing audio and video snippets of a person, remote channels like videoconferencing and voice calls, that may have been used for verification, are more easily infiltrated. We must work to find convenient alternative out-of-band channels that provide clear guarantees, and trusted introductions may prove a promising direction.

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

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

6 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 usability research [5].

We are also investigating the group messaging setting as an interesting avenue, which may enable more efficient introductions and provide insights about the trust relations between participants.

The direct verification of scanning a QR code is not hardened and can be faked. Strengthening this step is worthwhile and we envision that a ceremony akin to an NFC handshake may be designed to enhance security and usability.

Finally, the operating context on introductions goes far beyond what has been discussed in this paper. Additional variables like the user's view of the confidence levels, existence knowledge of other users, or existing message requests play an important role in a practical system. They form a landscape of varying perspectives for each user which influence both usability and the security guarantees we can provide. Formalizing these semantics more thoroughly would be beneficial to designing sound implementations.

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, François Wirz, Tsutomu Shimomura, and Kate O'Brien for their 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. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Green, M., Landau, S., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., Specter, M., Weitzner, D.J.: Keys Under Doormats: Mandating insecurity by requiring government access to all data and communications (Jul 2015), https://dspace.mit.edu/handle/1721.1/97690, accepted: 2015-07-07T02:15:02Z
3. Facebook: Messenger secret conversations, https://about.fb.com/wp-content/uploads/2016/07/messenger-secret-conversations-technical-whitepaper.pdf
4. 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
5. Gloor, C.: Trusted introductions for the signal private messenger, https://trusted-introductions.github.io/
6. Green, M.: What's the matter with pgp?, https://blog.cryptographyengineering.com/2014/08/13/whats-matter-with-pgp/
7. 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

8. Ian Levy, C.R.: Principles for a more informed exceptional access debate, https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate
9. jlund: Technology preview: Sealed sender for signal, https://signal.org/blog/sealed-sender/
10. Marlinspike, M.: Facebook messenger deploys signal protocol for end-to-end encryption, https://signal.org/blog/facebook-messenger/
11. Marlinspike, M.: Gpg and me, https://moxie.org/2015/02/24/gpg-and-me.html
12. Marlinspike, M.: Signal on the outside, signal on the inside, https://signal.org/blog/signal-inside-and-out/
13. Marlinspike, M.: Textsecure, now with 10 million more users, https://signal.org/blog/cyanogen-integration/
14. Marlinspike, M.: Whatsapp's signal protocol integration is now complete, https://signal.org/blog/whatsapp-complete/
15. Marlinspike, M.: The x3dh key agreement protocol, https://signal.org/docs/specifications/x3dh/#identity-binding
16. Parrelli, G., Rose, J., nightflame2, bitmold, Henthorne, C., Hart, A.: Android implementation security numbers, https://github.com/signalapp/Signal-Android/blob/main/app/src/main/java/org/thoughtcrime/securesms/verify/VerifyDisplayFragment.java
17. Perry, M.: [tor-talk] why the web of trust sucks, https://lists.torproject.org/pipermail/tor-talk/2013-September/030235.html
18. Rose, J.: Rust implementation fingerprint generation, https://github.com/signalapp/libsignal/blob/main/rust/protocol/src/fingerprint.rs#L154
19. 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
20. Signal: Signal technical information, https://signal.org/docs/
21. Tong, W., Gold, S., Gichohi, S., Roman, M., Frankle, J.: Why King George III Can Encrypt p. 13
22. 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
23. WhatsApp: Whatsapp encryption overview, https://faq.whatsapp.com/820124435853543
24. 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)
25. Zimmermann, P.: Why i wrote pgp, https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html