Incidental Incremental In-Band Fingerprint Verification: a Novel Authentication Ceremony for End-to-End Encrypted Messaging

Nathan Malkin
nmalkin@umiacs.umd.edu
University of Maryland

ABSTRACT

End-to-end encryption in popular messaging applications relies on centralized key servers. To keep these honest, users are supposed to meet in person and compare "fingerprints" of their public keys. Very few people do this, despite attempts to make this process more usable, making trust in the systems tenuous. To encourage broader adoption of verification behaviors, this paper proposes a new type of authentication ceremony, *incidental incremental in-band fingerprint verification* (I3FV), in which users periodically share with their friends photos or videos of themselves responding to simple visual or behavioral prompts ("challenges"). This strategy allows verification to be performed *incidentally* to normal user activities, *incrementally* over time, and *in-band* within the messaging application. By replacing a dedicated security task with a fun, already-widespread activity, I3FV has the potential to vastly increase the number of people verifying keys and therefore strengthen trust in encrypted messaging.

ACM Reference Format:
Nathan Malkin. 2022. Incidental Incremental In-Band Fingerprint Verification: a Novel Authentication Ceremony for End-to-End Encrypted Messaging. In *New Security Paradigms Workshop (NSPW '22), October 24–27, 2022, North Conway, NH, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3584318.3584326

1 INTRODUCTION

End-to-end encryption (E2EE) guarantees its users confidentiality and integrity of their communication by relying on public key cryptography. A critical challenge of deploying such systems is deciding how users discover each other's public keys. Applications like iMessage and WhatsApp have achieved massive success in usability and adoption of E2EE by utilizing a central server, operated by the service itself, which stores and manages all public keys for the service's users. A necessary property of this setup is that the key server is trusted; otherwise, it can lie to users when answering key lookups. Specifically, if Alice wants to send a message to Bob, she will ask the server for Bob's public key; the server can lie by providing a different public key—its own. Then, when Alice sends Bob the encrypted message, the server can intercept it, decrypt and read it, re-encrypt it with Bob's real key, then send it along. If this meddler-in-the-middle attack (MITM) occurred, Bob would be unable to detect it. Such attacks are an ever-present threat on currently deployed E2EE services due to the risk of server compromise as well as recurring demands from law enforcement agencies around the world, who are interested in weakening E2EE guarantees through techniques such as "ghost users" [9], despite the risks this entails [1].

One way to keep the server honest and prevent such attacks is for Alice and Bob to get together in person and make sure that what Alice believes to be Bob's public key really is his key (and vice versa). For example, Alice can read her version of the key out loud while Bob listens carefully for any differences. In practice, rather than comparing the full-length keys, it suffices to check shorter strings derived from the keys, known as fingerprints. Many programs that implement end-to-end encryption allow their users to perform fingerprint verification. One often-implemented improvement is allowing users to scan QR codes displayed on each other's phones, as a substitute for reading numbers out loud.

Even with this enhancement, few people perform fingerprint verification, for a number of reasons [17, 47, 48, 51]:

- Verification requires the two parties to be physically together, which is not always possible.
- Verification is not automatic: it requires time, effort, and focused attention from the user.
- Verification needs some motivation for the users. i.e., people want to understand why they are being asked to perform this ceremony. This is complicated by the fact that:
- Understanding verification requires some grasp of the concepts of end-to-end encryption. However, this is advanced knowledge, and applications have achieved success precisely because they do not need to explain these ideas to their users.
- People trust their service provider.
- People do not care enough about the privacy of their communication to incur the costs of verification.

The goal of the present work is to design a verification procedure that addresses some of these pain points and would be used by the general public. This paper proposes *incidental incremental in-band fingerprint verification* (I3FV), which features the following user experience: as part of their normal app usage, a user shares a selfie video or photo with their friends; similar to popular apps such as TikTok or Snapchat, the user's post includes a snippet of music, a visual "filter," or another type of prompt that the sender interacts with in some way (e.g., singing along to the song). Importantly, not all such "challenges" are available to *every* user each day—instead, their availability is determined by the user's key fingerprint. Therefore, which challenge someone completes reveals a small portion (e.g., one bit) of their fingerprint. To perform validation, recipients are asked a simple question to disambiguate which of

NSPW '22, October 24–27, 2022, North Conway, NH, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9866-4/22/10.
https://doi.org/10.1145/3584318.3584326

the day's challenges the sender completed. As this process repeats over time, the full fingerprint is gradually verified.

The core idea of this new process is that fingerprint verification can be: (1) incidental, (2) incremental, and (3) in-band. A crucial innovation of this design is that, instead of users explicitly engaging in the task of comparing fingerprints, the verification can take place largely *incidentally* to other actions the users perform. Concretely, users complete "challenges" by interacting with topical audio or visual features, analogous to those already popular on apps such as TikTok and Snapchat. Examples might include prompts the sender can respond to, songs they can lip-sync to, or augmented reality features that add objects or effects to captured images.

Another novel characteristic is that, rather than a one-shot interaction, fingerprint verification can take place *incrementally*, with the relying party gaining confidence over time that the key obtained from the server matches that of the recipient.

Finally, the challenges can be completed *in-band*, i.e., through the same app users need to verify keys for. While, theoretically, this would allow an attacker to subvert the verification process, the probability of being detected is high and is compounded at scale.

The remainder of this paper is dedicated to exploring the ideas of incidental incremental in-band fingerprint verification. It begins, in Section 2, by reviewing currently available trust models and the research that exists about them. Section 3 presents the concept of incremental verification in greater detail, with Section 4 outlining the resulting end-user experience. Section 5 analyzes the scheme's security properties, with Section 6 discussing additional deployment considerations. The proposed process offers a wide latitude of implementation details; Section 7 characterizes this design space, while Section 8 discusses how the concepts could be explained to end users. Finally, Section 9 discusses the open questions that this new model raises and potential avenues for investigating them.

I3FV represents a fundamentally new paradigm for obtaining trust for end-to-end encrypted communications. All previous approaches, both fingerprint verification and the even less popular Web of Trust methods, require users to dedicate themselves to the security task—which is one-shot, all-or-nothing. This is a fundamental limitation, as it requires users to have the needed knowledge and contribute the necessary time, effort, and attention; many years of deployments have shown that these are not forthcoming. In contrast, the I3FV approach only asks users to continue what they are already doing—sharing selfies and engaging each other creatively. It does not require any of the involved parties to know anything about security, nor do they even need to know the security reasons behind their tasks. As a result, I3FV significantly simplifies the process of key verification and has the potential to vastly increase the number of people who perform this security-critical task. If successful, this approach could keep widely deployed end-to-end encrypted systems more honest and may be applicable to other domains where trust is currently centralized.

2 BACKGROUND AND RELATED WORK

While end-to-end encryption is now widely deployed, many users lack a complete and accurate mental model of how it works [2, 3, 52]. One of the areas where gaps exist, though it may be possible to bridge them with proper explanations [6], is regarding the root of trust in these systems. Unger et al.'s SoK on secure messaging [44] provides a comprehensive overview of trust establishment approaches available to messaging applications. These include:

- Opportunistic (optional) encryption
- Trust-on-first-use
- Key fingerprint verification
- Short authentication strings (read out-loud during calls)
- Mandatory verification (fingerprint verification is *required* before communication)
- Authority-based trust (centralized, in which a central server is fully trusted, or decentralized, relying on Certificate Authorities and other elements of Public-Key Infrastructure)
- Transparency logs (e.g., CONIKS [30])
- Web of trust (such as that used by PGP [43])
- Posting keys publicly on established social networks (as in the commercial product Keybase [24] and some research prototypes [46])
- Identity-based cryptography
- Blockchains

All of these approaches have trade-offs, and most of them have seen limited deployment, except for fingerprint verification. If messaging platforms implement systems like CONIKS [30] or the recently-proposed Key Transparency with Anonymous Client Auditors [53], this may obviate the need for humans to verify key fingerprints. However, for the time being, this remains the only widely-supported trust establishment mechanism.

The problem of comparing hashes for authentication dates back decades and goes beyond secure messaging. While certificate authorities and associated public-key infrastructure allow trust to be bootstrapped for most data obtained over the Internet, there are no ways to verify that the root keys stored on devices are authentic. The proper solution, as pointed out by Perrig and Song in 1999 [33], would be to compare the fingerprint (i.e., hash) of a locally stored key with a reference, out-of-band: "*Since the user does not trust data downloaded from the network, the reference fingerprint needs to be passed over another channel, for example printed in a newspaper like the New York Times.*" Even if newspapers were in the business of publishing root keys, users would face the unexciting task of comparing 36 fingerprints, each consisting of 32 characters, letter-by-letter. Since "human limitations" would prevent most people from succeeding at, or even attempting, this task, Perrig and Song proposed an alternative: Random Art, a visual representation of the hash, which would allow users to compare images rather than text.

Since this first proposal, researchers have invented and evaluated a variety of new methods for performing this "authentication ceremony." Hsiao et al. [19] studied several different hash comparison techniques, using different types of images besides Random Art, as well as text with characters from different East Asian languages, which they evaluated with 400 participants, including some from Japan, Korea, and Taiwan. Kainda et al. [23] focused on verification *behavior* rather than modality, evaluating compare-and-confirm, compare-and-select (a string from multiple-choice options), and compare-and-enter (the hash into a text box to verify it).

More recently, Tan et al. [42] tested eight different textual and visual fingerprint representations, notably including auto-generated pictures of unicorns. Their study found that the representation can have a significant effect on the success of authentication: the best configuration allowed attacks to succeed 6% of the time; the worst 72%. Other researchers have focused specifically on text-based fingerprints, with Dechand et al. [12] finding that people are prone to errors and suggesting sentences as the best representation.

Visually comparing hash representations is not the only way of performing fingerprint verification. In 2005, McCune et al. [29] proposed using the camera on one phone to scan the screen of another one. Today, essentially all encrypted messaging applications allow users to verify keys by scanning QR codes on each other's phones.

Because authentication ceremonies are now built into encrypted messaging apps, researchers have been able to study them in situ. Vaziripour et al. [47] performed a user study of fingerprint verification in three apps: WhatsApp, Viber, and Facebook Messenger. They found gaps in user understanding and awareness, as well as discoverability issues that resulted in success rates that were as low as 14% in some cases. Due to the various usability issues uncovered, Vaziripour et al. [48] and Wu et al. [51] sought to redesign in-app messaging and explanations in Signal's authentication ceremony; while they achieved increased understanding, many participants were still confused about the concepts involved. Most recently, Fassl et al. [14] tried to improve the verification process through user-centered design, including collaborative design workshops, selecting viable candidates, iterative storyboard prototyping, and a mixed-methods online evaluation. Despite these efforts, they noted that "the quantitative comparison of our prototypes did not reveal usability or user experience improvements." Similar outcomes across various research led Herzberg et al. [17] to conclude that "secure messaging authentication ceremonies are broken."

Though not widely adopted, alternatives to existing authentication ceremonies have been proposed; many of them fall under the umbrella of the more general problem of secure device pairing [15]. Besides visual-based techniques, there are also audio-based approaches [16], which researchers have compared against alternatives in user studies [25, 26]. Other novel approaches include shaking devices to securely pair them [28, 41] and exploiting correlated magnetometer readings for authentication [22]. Such techniques generally require the two devices to be near each other.

A technique that does not require physical proximity is "Short Authenticated Strings" [45]. In it, users read words, generated from their keys, out loud during an audio call. This was implemented, for example, in the ZRTP VoIP protocol [54]. In addition to being a distraction from the primary reason for the call, this method relies on the parties knowing each other's voices and being able to recognize attacks through imitation, which creates some security risks [34, 36]. Nonetheless, this method directly inspires the I3FV approach. I3FV also shares some features with the work of Dabbour and Somayaji [11], who investigated the possibility of in-band authentication by seeing whether people can distinguish friends from attackers entirely through their texting style.

By relying on the attentiveness of human contacts for verification, I3FV shares similarity with social authentication, which is another trust establishment method [4]. Proposed primarily as a solution for last-resort authentication [20] (e.g., recovering a lost password), the core idea is for a user to designate several "trusted contacts" who have to verify the user's identity in order for account access to be restored [7]. While this scheme can be resilient to social engineering [35], it has seen only limited deployment [13], and user studies have uncovered drawbacks in its convenience and efficiency [27]. However, in these prior versions of social authentication, the security task (i.e., the authentication ceremony) is the primary activity, and it typically must be performed out-of-band [37]. In contrast, I3FV aims to do away with these limitations.

3 IDEA DEVELOPMENT

This section describes the rationale and advantages of I3FV by examining design goals, alternate approaches, and detailing the final design.

3.1 Design considerations

The novel verification scheme proposed in this paper has two main goals. The primary one is to maximize adoption of verification behaviors. Counter-intuitively, this may come at the cost of efficiency: the current solution—scanning QR codes—is already quite efficient, but people lack the motivation to do so. The second requirement is maintaining compatibility with the architecture of existing E2EE messaging apps, to ensure the solution can actually be deployed in currently popular products. This entails maintaining the reliance on central key servers for key retrieval and contact discovery, despite the limitations of this model.

There are also two *non-goals*—objectives the design does not aim to meet. The solution does not need to be more secure than in-person fingerprint comparisons, since those are already optimal. Instead, trading off some amount of security for adoptability is acceptable. Additionally, it is not a goal for this scheme to *always* be used by *all* people—but rather that any person *could* use it, resulting in more people performing fingerprint verification than today. As a consequence of these properties, users who require greater security—for example those who are at a greater risk of targeted attacks—would be better off continuing to use traditional authentication ceremonies. (In practice, however, at-risk users often face extraordinary time and resource constraints [49], so they may benefit from *incidental* authentication.)

With these goals as a starting point, the following requirements guide the design:

1. Users should not need to be physically present to perform verification. For many, meeting up in-person is not feasible.
2. The verification process should not prevent, interrupt, or delay normal communication through the app. These actions would hurt usability, annoy users, and hinder adoption.
3. Users should not need to understand encryption or other cryptographic concepts, as this sets too high a barrier for participation.
4. Users should want to perform the verification process even if they do not prioritize privacy, since many do not. Ideally, the verification process would be intrinsically motivating.

3.2 Design development

Let's begin by considering some strawman approaches, before arriving at the final design.

Approach 1. Users could read their fingerprints (or an even more shortened version of the key) at the beginning of a voice or video call. This provides authentication, assuming voice and video are hard for an attacker to spoof. It also addresses the first requirement, since the two parties no longer need to be physically colocated. As mentioned in Section 2, the ZRTP protocol used this approach for encrypted VoIP [54]. In practice, many people do not perform this ceremony. It contradicts normal phone etiquette (we expect to start calls with greetings), distracts from the main purpose of the call, and parties may not be concerned enough about security.

Approach 2. What if, instead of reading the Short Authentication String at the beginning of calls, each user would record a video of themselves saying it? This video could be made available to the user's contacts, who could verify it on their own time. This approach would be more acceptable because it would no longer interrupt normal conversations. Still, there are some problems. Chief among them is the question of motivation. Why would people record these videos? And why would their friends take the time to watch them? If greater security is the only reason for doing so, many will not bother.

A novel but growing concern is how easy it would be to fake these videos. With improvements to computer graphics and speech synthesis, it has become possible to manipulate existing footage and recordings to generate completely new recordings of people. In essence, you can make anyone say anything. This "deepfake" technology is nascent, but is available and improving [31].

Inspiration. Many people already record short videos and photos of themselves, which their friends can watch at their convenience. Popularized by Snapchat, similar features are also offered as Instagram Stories, Facebook Messenger Stories, WhatsApp Status, and Signal Stories [38]. Though typically free-form, apps allow users to customize them with overlays (referencing current events, location, time of day, etc.), stickers, and text. In many cases, users may alter how or what they record based on the planned overlay. These customizations make each new photo or video distinct and therefore harder to fake. In fact, some apps already require users to customize selfies to deter fakes. For example, users who wish to verify their profile on the dating app Bumble are "prompted with an example of one of a hundred random photo poses"; they are then instructed to "take a selfie mimicking that pose" [8].

3.3 Proposed principles

The I3FV idea has a few key components:

Actions over words. Rather than reading numbers or words out loud, users can record videos of themselves performing different actions. Like the number or word, the action is deterministically chosen based on the user's key. Thus, the receiving party (or, more specifically, their app) can figure out which action the sender was supposed to be performing, knowing just their key.

Actions, or "challenges," are more interesting for users to perform. Even basic gestures ("fist-bump someone or something") can be more fun than saying words out loud. Actions are also likely harder for potential attackers to fake. The richer the movement, context, and interactions in the video, the more effort it requires for an attacker to reproduce or alter.

Constrained but creative. The more engaging and entertaining a challenge is, the more likely someone is to complete it and then come back to perform another one. Challenges should therefore engage users' creativity and offer them multiple ways to go about it. This is subject to one primary constraint: the recipient of a challenge video should be able to confirm to their app that the sender performed the intended action. Thus, completely open-ended challenges ("do something unexpected") will not work, but more constrained ones can be good candidates ("perform a dance move"), as long as there are no similar challenges that they can be confused with ("pretend you're in a ballet").

More and shorter, not one longer video. To securely represent a fingerprint as a single challenge video, either the number of potential challenges would have to be very high, or the user would have to perform a series of challenges back-to-back. I3FV is based on the hypothesis that, instead of a single long video, people would prefer recording multiple shorter ones, over a longer time span. This has a number of advantages:

(1) Each video can be dedicated to a single challenge.
(2) Users can choose when they want to record videos and which challenges they want to complete.
(3) Verifiers only need to watch one short video at a time, which reduces the effort and commitment required.
(4) Whereas staking verification on a single video would give attackers a single target, multiple different videos requires repeated and ongoing effort from attackers.

App integration. I3FV could be a feature added to existing apps like Snapchat. When users go to take a photo or record a video, they have the option of viewing the day's challenge. (The challenge is derived from their key and today's date, and thus is different from many of their friends' challenges.) They can complete it, if they so wish, by performing the stated actions.

As with all other photos/videos in the app, a challenge video can be sent to specific friends directly, or passively posted for any friends to see if they click on the user's profile. (Note that, with all apps, there is an indicator when one of your friends has new content available, so a friend would not need to guess whether a video is available.) A friend watching the challenge video could confirm to their app that the user performed the correct challenge by answering a simple question (yes/no, or multiple choice).

4 USER EXPERIENCE, DETAILED

The final user experience of I3FV may resemble the following.

A user opens their messaging app and decides that they want to share a photo or video of themselves with their friends. Importantly, their motivation for doing so is sharing content with their friends; any security properties of what they are about to undertake will happen *incidentally*.

When the user opens their app's camera mode, they can swipe (or otherwise interact with the screen) to see which customizations they can apply. For example, swiping once may overlay a banner image celebrating today's holiday ("National Honey Day!"); swiping a second time may turn on an augmented reality lens that transforms the person's face to look more like a bee (see Figure 1); and so on. Apps such as Snapchat already provide these customization options,

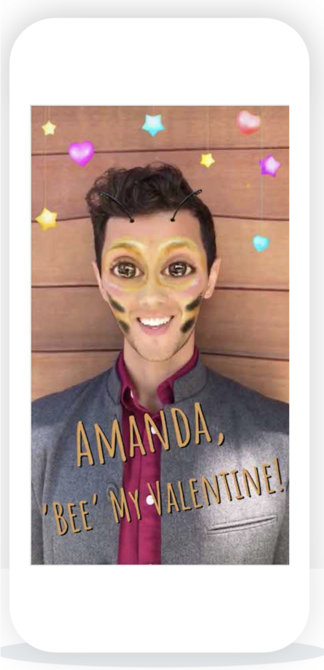


Figure 1: A Snapchat lens (source: snapchat.com) showing a sample face manipulation. To make it an I3FV challenge, this lens would need to be available for a subset of the user base, based on their public key.

and, again, the user's motivation for choosing a specific lens or overlay is just whatever they consider most fun and entertaining.

However, possibly unbeknownst to the user, not all possible lens options were available to them when they were swiping through the options. The "bee" lens was only shown because the user's public key ended with a 1.

Once the user records themselves using the lens and posts the photo or video, it becomes available to their friends for viewing, on their own time. This is fully analogous to how many current apps have a "Stories" feature that allows users to view their social connections' posts until those disappear after 24 hours.

An I3FV implementation changes the user experience of viewing others' posts in only one small way. Transmitted with each post would be a piece of metadata, which indicates whether this post represents a verification challenge. If it is, the recipient's app can use its version of the sender's public key to compute which challenge it expects that post to be. Then, when the user has viewed the post, the app can prompt them with a simple question, for example, "which animal did the sender most resemble: alpaca, bee, cassowary, or dingo?" If the user's answer matches the expected value, one bit of verification is complete. This aspect of the user experience is new and not incidental to everyday usage—fingerprint verification is the only reason for its existence. However, it is low-effort (so as to be minimally annoying) and happens automatically, so the user does not need to remember to make a security decision.

5 SECURITY ANALYSIS

This section analyzes the security aspects of I3FV compared with existing fingerprint verification techniques.

5.1 Threat model

This analysis assumes the same threat model used by end-to-end encrypted messaging apps. Namely, the server can perform active attacks on key lookups and the communication itself, if it has the necessary keys, since all content passes through the server. Clients (i.e., the messaging apps installed on end-user devices) are assumed to be trusted, with each one having a distinct public key.

Caveats about trusting clients. In today's real-world deployments, client apps are developed and distributed by the same entities that operate the servers. Thus, if a service turns evil, they could distribute a malicious version of the application, rather than performing any machinations on the server. This is a fundamental limitation of today's deployments that I3FV does not address. Traditional fingerprint verification also suffers from the same problem. However, there are classes of attackers that can only target the server; adding back-doors to apps is generally costlier and more difficult.

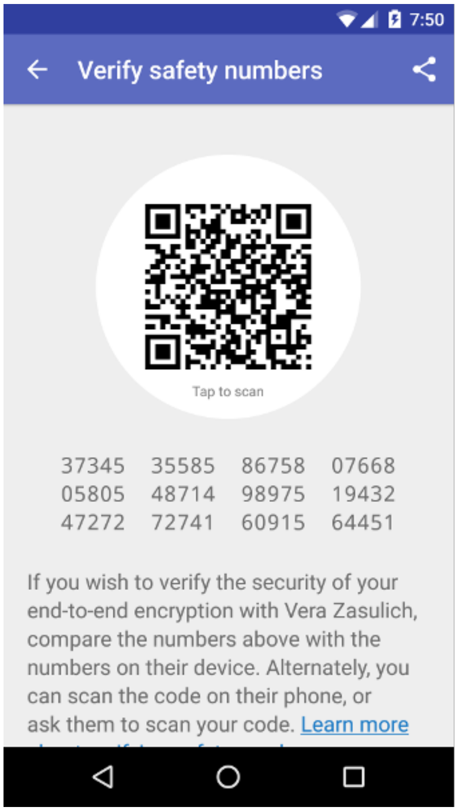
Clients can also become untrusted if the user installs malware on their phone. If that malware compromises the phone's operating system, then nothing is safe anymore. If privilege escalation does *not* happen, a locally running app or a network attacker may still be able to learn some things about the messaging app and the user's actions through side channels. Such attacks are out of scope.

Mass surveillance vs targeted attacks. Under E2EE's strong threat model assumptions, the MITM attacker may have a variety of capabilities; however, in practice, any attack comes at a cost. As discussed in Subsection 3.1, the goal is not to replace one-shot fingerprint verification for those who need or want to do it, but to engage more people in the practice. Even if each individual achieves a lower level of security overall, the (hopefully) large scale at which I3FV can be adopted significantly increases the possibility that a misbehaving server is caught, thus raising the risk and cost to an attacker. Such friction is especially important to thwarting mass surveillance (a priority that has been identified by popular apps such as Signal [50]), which relies on MITMing a population as a whole, rather than targeting individuals.

5.2 Entropy

Traditional verification authenticates the entire fingerprint at once, while I3FV validates only a few bits at a time. How long does it take to verify the entire fingerprint using the new approach?

Signal and WhatsApp's "safety" and "security" numbers consist of 60 (base-10) digits (see Figure 2). Because the safety number is a combination of the two parties' keys, one person's key accounts for half of those, or 30 digits. Correspondingly, the number of possible fingerprints is 10^{30}. If 10 challenges are available each day, it will take 30 days to verify a fingerprint with equivalent entropy. (Verification need not happen back-to-back, so those 30 days can be spread out according to the sender's convenience.) If fewer daily challenges are available, e.g., just four, verification will take somewhat longer: $\log_4(10^{30}) \approx 50$ days. At a maximum, if only two challenges are available every day, then verification will



37345 35585 86758 07668
05805 48714 98975 19432
47272 72741 60915 64451

If you wish to verify the security of your end-to-end encryption with Vera Zasulich, compare the numbers above with the numbers on their device. Alternately, you can scan the code on their phone, or ask them to scan your code. Learn more

Figure 2: Signal's screen for verifying a safety number

take $\log_2(10^{30}) \approx 100$ days. Where in that range a service lands is a tunable parameter; its main determinant is a service's capacity for generating unique challenges, a question that is discussed shortly.

These calculations represent an upper bound for how long verification can take. In practice, high levels of assurance can be achieved with fewer challenges because, as more and more bits of a key are verified, the probability that only the remaining unverified bits have been spoofed becomes increasingly lower. (The exact likelihood depends on the structure of the key space and the ease of finding near-collisions.) Furthermore, even a small number of failures might be strongly diagnostic of the presence of attackers and could trigger more explicit full verification on that basis.

5.3 Blocking and replay/ordering attacks

One of the distinct aspects of I3FV is that verification is done in-band. Strictly speaking, this is not necessary: a user could perform verification out-of-band, across different services, for example by posting challenges verifying their Signal keys to Instagram. Realistically, though, the need to support both sender and recipient actions makes cross-app verification cumbersome and unlikely to be implemented. If verification *is* in-band, then the challenge recordings are being sent through a potentially malicious server, therefore presenting it with an opportunity to tamper with them.

One action a malicious server may take is blocking messages containing challenges from being delivered altogether. Such a denial of service attack could effectively prevent verifications from happening. However, implementations of I3FV could mitigate this by taking advantage of "read receipts": these features are present in most messengers and social media applications and allow a sender to see when their messages have been delivered or read. If users notice that none of their completed challenges are being viewed, this might arouse suspicions. Even without this feature, this attack would break down at scale, if the user base were to notice—through conversations and anecdotal observations—that their messages failed to be delivered. Since the goal, as discussed previously, is to thwart mass surveillance, this may offer a reasonable degree of protection.

If a service utilizing I3FV reuses challenges, an adversary may attempt to take advantage of this by significantly delaying verification messages. Ideally, each day would bring a new, unique set of challenges; however, these require effort to come up with and encode in the app. This effort is furthermore continual, as the scheme expects the service provider to keep issuing new challenges indefinitely into the future. To simplify their work, service providers may therefore find it compelling to reuse challenges. However, this opens up an avenue of attack: if a user completes one challenge, but the MITM attacker's key is associated with a different one, an attacker might withhold the delivery of the user's recording, and then send it at some point in the future when that challenge *is* appropriate for the MITMed key. (They may similarly attempt to replay already-delivered recordings, in the hope that the recipient does not spot the repeat.)

There a few options for dealing with this attack. One possibility is to not provide any special defenses and leave the protocol as-is. This is not unreasonable, as users may be expected to flag occurrences of reordering because they are naturally suspicious: if a message is delivered much later than it is recorded, the sender may now have a different haircut or physical surroundings, and some recipients are likely to pick up on that. To defend against the attack more directly, apps could ask senders to somehow mention the date (or current events) when challenges are repeated; this may make the content less natural, but would solve the problem. A way to address the issue without direct user involvement would be to incorporate a timestamped watermark into the recording, either overtly or through cryptographic means [10].

5.4 Impersonation

In addition to interfering with verification by blocking or reordering messages, an attacker might try to tamper with their contents. The interference can come in two ways: (1) modifying verification messages or (2) creating new ones altogether, in which the sender is being impersonated. The success of these attacks depends on two intertwined factors: the susceptibility of the verification challenges to alteration (referred to, going forward, to as "forgeability"), and the effectiveness of technical tools for creating deepfakes or otherwise impersonating an individual.

The forgeability of challenges is a spectrum of difficulty. On the easier side are basic visual overlays ("filters" or "stickers"); these can vary from a border around the edges of the photo or video to more complex images that cover up only parts of the frame, with varying opacity levels. An attacker wishing to attack a filter-based challenge could possibly overlay one filter with a different one, or attempt to remove or otherwise alter the overlay.

This possibility need not eliminate filters as a challenge category. People might (and should be encouraged to) engage with them in creative ways, such as making a face, pointing, or incorporating them into their pose. Users might also refer to them in a caption, which occludes some of the image. All of these will increase the complexity of altering the image, if slightly. Most simply, filters could spark a conversation between sender and recipient. To succeed, an attacker would therefore need to not only carefully consider and modify the original message in a custom way, but would have to monitor and potentially act on the entire conversational context. This very likely could not be done in an automated way, requiring individualized human attention. Thus, while modifying filters could be plausible as a targeted attack, doing so at scale, as necessitated by mass surveillance, would be prohibitive.

This speaks to an important dimension of the forgeability of challenges, which is how much engagement they generate with the user. As was just discussed, post-hoc interaction (i.e., starting a conversation between creator and verifier) is good for security, since an attacker risks discovery if it is not managed. Still, the primary dimension of engagement is the degree of involvement in the challenge by the sender. If they completely ignore the content of the challenge and take a regular selfie, or even a photo without themselves pictured, this represents a total lack of engagement. In contrast, a maximally engaging challenge would result in content that the sender would create only for this challenge and under no other circumstances, so that it cannot be used in other contexts.

The reason engagement is important for security is due to the threat of manipulated or synthetic photos or videos, especially ones created using machine-learning based techniques (deepfakes). Since they first became known in 2017, deepfakes have been widely created and studied [31], becoming increasingly complex and believable with continued advancements in the technology. Moreover, similar techniques have made it possible to synthesize fake speech as well [21]. Current deepfakes are subject to limitations, such as struggling to handle non-frontal poses [5] (hence the emphasis above on full-body motion in I3FV challenges). Unfortunately, these constraints are likely to be overcome with time.

Along with improvements in creating deepfakes, research has also produced advancements in detecting them [31]. These too continue to evolve and could eventually be built into messaging apps (which are trusted under the E2EE threat model) to scan incoming challenges automatically and flag suspicious occurrences.

Another defensive strategy apps can use is varying the influence of different verification types. For example, if a certain challenge is harder to spoof (e.g., a video versus a photo, or an activity that requires engaging with the surroundings versus standing still), it could count as a higher-certainty verification. Fewer of these may be required to reach the goal of full key verification.

The coming years will undoubtedly see improvements in deepfakes, and, if the technology is perfected, it would threaten the reliability of in-band verification. Nonetheless, I3FV may remain viable in the foreseeable future. For deepfakes to completely negate this verification method, first, there would need to be sufficient audio and video and data to synthesize them for every person using the platform. The synthesis task becomes even harder if the actor needs to interact with the environment. The attacker would need to ensure convincing surroundings, which requires knowing where the sender is and providing a plausible and dynamic background, including features such as weather. Similarly, they might get caught if someone notices that the target's clothes do not match what they were wearing in other videos that day. There might also be subtle communication cues that give the attacker away [11]. Since many people will be performing the same challenge, another cause for suspicion would be if all the synthetic content looked too similar, for example if motions and gestures in different videos all looked the same. Finally, all of these deepfakes would need to be generated on-demand and in near real-time, since users will notice if there is a delay in reaching their audience. After all that, if the attacker succeeded in one particular instance, they would need to do this repeatedly, day in and day out, without arousing suspicion, as long as the user was continuing to perform challenges. They would also need to overcome potential countermeasures, as discussed above.

Accounting for all this complexity, it might be still be possible for a targeted attack to succeed, in select circumstances, though it would not be easy. But, at scale, when even a small amount of detections can raise red flags and lead to investigations of the server, pulling this off may be cost-prohibitive. Recall that the aim (outlined in Subsection 3.1) is not to fully eliminate the possibility of attacks or replace traditional verification, but to thwart mass surveillance by getting more people engaged in checking the server, so if something suspicious does happen, they can revert to traditional fingerprint comparisons. Therefore, while impersonation certainly presents a challenge to I3FV, it does not preclude the system meeting its goals if deployed in a real, widely-used service.

6 DEPLOYMENT CONSIDERATIONS

Beyond any theoretical security properties, the success of a system depends on whether it is implemented, used, and supported.

6.1 Adoption incentives

Whether a system gets deployed is driven in large part by the incentives surrounding its adoption.

Platform adoption. Why would a platform adopt incremental verification? After all, doing so would harm their ability to carry out MITM attacks, which they may want to perform due to governmental pressure or commercial advantage. However, the same exact pressures, coupled with control of both the clients and the server, weigh against a platform's decision to adopt E2EE in the first place. Yet many still do.

Behind such decisions are a variety of factors. Platforms want to win users' trust and demonstrate that they are on the side of users and their privacy. Ensuring that data is encrypted also limits the ability of hackers to access it. Many also do not want to deal with government requests, since complying with them requires considerable resources, especially if they need to be handled across many jurisdictions. Compliance may also lead platforms to suffer reputational damage, if the handing over of private user data is publicized. All of these pressures that incentivize offering E2EE may also lead to the adoption of I3FV.

Furthermore, if some messaging platforms adopt I3FV, this could create competitive pressure on others to do the same. This is particularly true if completing the verification challenges turns out to be—as hypothesized—fun and self-motivating; then, implementing this feature would hold the promise of an increased user base for those who choose to adopt this technology.

Platform support. Deploying incremental verification carries more than a one-time cost; the scheme requires ongoing upkeep and support. One specific limitation that may affect the deployment of I3FV to real systems is the cost of curating challenges. A service may require hundreds of challenges, and while reuse is possible (as discussed in Subsection 5.3), having fresh ones would be better. This is inherent to the method, but it need not deter adoption. First, ideas can be crowdsourced; for example, Snapchat already allows users to contribute filters [39]. Second, many E2EE services, such as WhatsApp and iMessage, are operated by profitable entities (Meta/Facebook and Apple are, as of 2022, both valued at hundreds of billions of dollars or more [32]), which are likely able to afford to spend a small amount of money to generate challenge ideas.

As with the choice of initial adoption of I3FV, the incentives behind funding ongoing support of this feature may include competitive demands, reputational pressure, and the increased user engagement that these features may drive for the app. In contrast, if the scheme turns out to have very few users in practice, then it is likely also not meeting its security goals. In this particular way, therefore, the incentives of apps and users may even be aligned.

6.2 Limitations

In addition to the security properties that make I3FV different from traditional fingerprint verification, this method has a set of particular challenges and limitations, which are worth considering.

First and foremost, I3FV is dependent on people completing challenges, i.e., sending photos and videos of themselves based on specific prompts. Whether this will happen depends on two questions: will users send photos and videos of themselves? And will they alter them in response to "challenges?" The ideal way to study these questions is empirically, but intuition suggests affirmative answers to both. Social media applications like Instagram, Snapchat, and TikTok have amassed hundreds of millions of users. While they host a variety of content, and exact data about it may not be publicly available, it is commonly understood that photos or videos of the users themselves represent a significant fraction of content being shared. Similarly, the viral spread of memes on TikTok and the popularity of filters on Snapchat (which the platform even targets for advertising [40]) suggest the willingness of users to get cues about the content to create from the platform itself.

Section 3.3 emphasized the need for challenges to be creative and engaging to minimize the risk of impersonation. However, this involves a certain level of trade-offs: challenges that are more involved, potentially embarrassing, or complicated ("stand on one leg while juggling tea-cups") will be completed by fewer users.

There will certainly be users of any service who are uncomfortable sending selfies, recording videos of themselves, or performing challenges. They will therefore be unable to take advantage of I3FV. While this is unfortunate, the scheme's goal is to increase how frequently authentication is performed across the entire population; it is not expected that every single user can or will engage in it. For such users, E2EE services should continue to provide the option to perform fingerprint verification using traditional means; I3FV ought to be strictly additive. In the future, researchers and developers may come up with additional authentication ceremonies, which—following the paradigm proposed here—may be incidental, incremental, and/or in-band. These methods may also be needed for E2EE services for which the light-hearted nature of I3FV is not a good fit, such as business messaging and professional social networks, or anonymous communication tools for whistleblowers, activists, and other at-risk users.

Another clear limitation of I3FV is that it takes a long time, as estimated in Subsection 5.2. This means it takes longer to reach the same security level as traditional verification, and also increases the risk of abandonment if users stop performing verification challenges. These limitations are inherent to the method, but any amount of verification is better than no authentication, which is the current reality of E2EE messaging for nearly all users.

Theoretically, a challenge may go viral and result in people completing it even though it does not match their key for the day. If the server is well-behaved, this should not cause problems, as the sending app would mark it as a regular post, and the recipient would not be asked verification questions about it. In an adversarial scenario, however, an attacker might attempt to leverage this post to verify their (MITMed) key. This has low probability of happening, as the video that went viral would have to match the challenge for the attacker's key. Moreover, this would only affect a few bits of the overall fingerprint; success would require this occurrence to repeat many times over, which is implausible.

In summary, while there are clear limitations, they may not necessarily affect the overall viability of I3FV as a new and potentially promising authentication ceremony.

7 DESIGN SPACE EXPLORATION

I3FV is a general paradigm, and any specific realization of it will require the implementers to make a number of decisions about its details. The aim of this section is to characterize this design space.

7.1 Types of challenges

For the purpose of I3FV, a *challenge* is some way to customize a photo or video of the sender that distinguishes it, in a human-verifiable manner, from others they could have made. There are a variety of options surrounding what challenges look like.

One simple but consequential design choice is whether verification messages are photos or videos. Videos are harder to fake and thus better for security. There are also more opportunities for creative expression in a video. On the other hand, photos may allow for more careful and deliberate self-presentation, which is an important goal for social media users. Ultimately, either is valid, and ideally services would provide options for both photos and videos. Audio-only challenges—including videos that do not show the face of the speaker—are also an option, but they rely on recipients being able to distinguish the sender's voice, while also being highly vulnerable to synthesis attacks [36]; they are therefore less secure overall and less ideal as a medium.

Type	Example
Visual overlays	A picture of a donut taking up a portion of the screen. If sufficiently inspired, the sender could "interact" with it by pretending to take a bite out of it.
Lip-sync to music	A 10-second clip of some song or audio clip that the sender sings or dances along to
Perform specific gestures	"It's Tongue-Out-Tuesday! Stick your tongue out!"
Act out charades	"Pretend you're a penguin"
Respond to specific questions	"What is your favorite ice cream flavor?"
Respond to open-ended prompts	"Talk about the best vacation you've ever had"
Use augmented reality lenses to modify face	The sender's face is manipulated to look more like a bee (see Figure 1)
Interact with characters and objects projected using augmented reality	An AR dinosaur is seen walking behind the sender, while they express fear.

Table 1: Categories and examples of challenges

The biggest question is what the specific challenges should be. Table 1 lists potential categories for challenges as well as specific samples. Examples include visual overlays ("filters"), lip-syncing, movements such as gestures or miming, monologues on provided topics, as well as interactions in the space of augmented reality. Beyond those collected here, there are more categories and of course many more examples of each type of challenge.

In addition to a user's key, other factors can help determine which challenges they get to select from. For example, similar to how filters work in today's popular services, challenges could be location-specific ("You're in New York City! Pretend you're eating a Big Apple!") or time-specific ("It's midnight and a full moon! Pretend to howl like a wolf!"). Any of the currently available customizations ought to work as challenges for I3FV, as long as their availability is *also* somehow influenced by the key hash (e.g., there are two versions of each challenge, or it is only offered to a subset of users at a given time).

Rather than utilizing all known challenge types, services may decide to make only certain types of challenges available, based on the character of the app. If filters are popular on Snapchat, they can retain them and simply add verification questions for some of them. If TikTok decides to use I3FV, they can focus on musically-inspired challenges. A hypothetical dating platform could have challenges where users share something about their hobbies and interests.

7.2 Mechanisms for verification

The discussion so far has focused primarily on senders and the challenges they must complete; the other half of the puzzle is the recipient: it is incumbent on them to verify the completed challenges. What exactly does this verification process entail?

The verification step should consist of a simple question asked after a recipient has viewed a contact's verification video. This can take the form of a yes/no question about the challenge being completed ("Did this video feature your friend pretending to be a bat?") or choosing one of several answers ("Which of the following animals was your friend pretending to be?").

If the recipient's answer matches the intended challenge, then a portion of the key can be considered successfully verified. But what if there is a mismatch? This could be an indicator of compromise, though it is generally more probable that it is a mistake either on the part of the sender or recipient. What are the next steps?

A simple solution is to display a warning about the mismatch to the user, urging them to use caution, perform traditional in-person fingerprint verification, and possibly cease communication until this is done. However, any such messaging would need to be carefully designed to account for the fact that false negatives (accidental mismatches) are incredibly more likely to occur than true negatives (the server actually being compromised). Because of this, some implementations may even choose to delay user notification until there's a greater confidence in a mismatch (i.e., several failed verifications). A fully informed solution for this may require data from a real deployment about how often mismatches occur in real life.

Whenever the app chooses to inform a user about a mismatch may be the first time they find out about the security purpose of the challenges: until then, implementations may choose to withhold the security rationale behind challenges and present them simply as a fun and playful feature. At this point, explaining the problem and the context may be challenging, as users are well known to struggle with mental models of end-to-end encryption [2, 52]. Effective explanations of I3FV and encryption in general may be topics for future work, but Section 8 presents an attempt at a metaphor aimed at explaining I3FV to a non-expert audience.

7.3 Incentivizing engagement

The success of incremental verification depends on both sides—senders and recipients—engaging in it, repeatedly, for long periods of time. What would motivate them to do so? One answer is that—hopefully—they consider it fun: the design of I3FV's method is based on the observation that people already perform most of the actions they need to do for challenges. While specific social media apps may come and go, the core behavior—sharing photos and videos with friends and family—is unlikely to ever go away.

The least invasive way to implement I3FV is to make the challenges available alongside similar customization features (e.g., filters) and let users discover this feature for themselves. To increase adoption, apps may choose to add further features to incentivize participation. For example, they may use streaks [18] and other gamification features to motivate users to complete verification challenges. Other examples might include badges, levels, and unlocking sticker packs or other features once a certain number of verifications has been achieved.

In general, while senders in I3FV bear the highest user burden—since they need to record challenges—the intent of the design is that there is no *additional* burden on them. If they are already creating and customizing photos and videos of themselves, then they only need to continue what they have been doing. If they are *not* doing this, then, as discussed in Subsection 6.2, I3FV may not be a good match for them, though perhaps some can be convinced through the gamification features or by observing their friends do it.

In comparison, recipients need to expend less effort—because all they need to do is answer simple questions—but this effort is entirely unique to the I3FV user experience. Arguably, recipients may require less incentivization to answer verification questions, since they are already likely to watch videos from their friends, and verification questions can be displayed automatically at the end. However, here too apps may choose to award points or otherwise gamify the interactions. Such inducements need to be carefully considered, however, as poorly designed incentives might lead people to claim someone correctly performed a challenge—even if they did not—in order to collect the promised rewards.

For both senders and recipients, an open question about the design is whether to inform users about the security motivations of the tasks they are performing. All of the approaches suggested above can be implemented without revealing this rationale. This may be preferable from the perspective of simplicity, so as not to burden users with confusing explanations about the mechanics of end-to-end encryption.

On the other hand, without some context, users may wonder why they do not have the same challenges available as their friends. More prominently, those on the receiving side may want to know why they are being asked to answer questions about their friends' videos. Explaining that this is being done to help verify their identity—without necessarily getting into the technical details—may be necessary to avoid user confusion.

Furthermore, providing the security background behind the actions might provide additional motivation, making some users more willing to perform challenges and verifications. It also unlocks other potential gamification and user interface options, such as showing the percentage of the key that has been verified.

Crucially, exposing the security motivation behind the actions does not negate any of the advantages I3FV has over traditional authentication ceremonies. In particular, the primary task of sending and watching videos is not being done for the purposes of security, but because this is how people want to interact with their friends.

The various design decisions discussed in this section are all valid implementation options for I3FV. Determining the best solutions is an open research question, which is discussed in Section 9.

8 EXPLAINING I3FV TO END USERS

I3FV relies on senders and receivers performing tasks that are already largely part of their usage patterns; therefore individual end users do not need to understand the logic and purpose behind the tasks they are performing. Nonetheless, there may be times when explaining the process to non-experts may be helpful or necessary. For such situations, this section presents a non-technical metaphor, which could be the basis for explaining I3FV to people who are unfamiliar with the concepts of end-to-end encryption.

> Alice and Bob are two friends who want to exchange letters and photographs over snail-mail. The postal service will reliably deliver any letters (and attachments) to the address on the envelope without opening it. There's just one problem: Alice doesn't know Bob's address.
>
> Luckily, there's a directory that lists everyone's address, so Alice can just check that. However, Alice is worried that the directory's publisher, Eve, may replace Bob's real address with her own. Then she'll get any mail meant for Bob, read it, and forward it on to Bob, who won't know that his mail has been intercepted.
>
> Alice would like to ascertain Bob's address, but she can't just call him and ask because she doesn't have his phone number either. Instead, she needs to continue communicating through the mail (*in-band*).
>
> There are some things Alice can try to do, which won't work. She can write her return address on the envelope, but Eve can just put the message in a new envelope. Bob can write his real address in the return letter, but Eve opens those letters too, so she can change it when the letter passes by her on the way back.
>
> But while Eve can alter written letters, she can't forge photographs, at least not well. So Alice and Bob come up with a solution: Bob will send a photo of himself that includes his address. Their first idea is for him to take a photo with a piece of paper with his address, but Eve might still be able to alter that portion of the photo, since it's just text.
>
> Instead, Bob will give Alice a photo-tour of his neighborhood (which, *incidentally*, he was already planning to do!). First, he'll share a photo of himself standing in front of his house, so the house number is visible. Next, he'll send a selfie with the street sign at the end of his block. And finally, there will be a photo of him with the "Welcome to ..." sign at the city limits.
>
> Each of these photos will feature Bob, so Eve won't be able to swap them out for ones with a different house or street sign. Therefore, with each new photo, Alice will *incrementally* gain confidence that she's sending letters to Bob's real address.

9 OPEN QUESTIONS AND RESEARCH DIRECTIONS

The biggest question about the proposed verification scheme is: will it work? Will people complete and verify challenges as required by I3FV? To answer this, it may help to know what fraction of the population, or of a given app's user base, already engages in behaviors similar to those required for I3FV (i.e., sends selfies). Another related research question is what fraction of a population actually *needs* to perform verification to catch a misbehaving server.

There are a variety of other more specific research questions that arise from this scheme. Many of them revolve around challenges specifically, since the choice of challenges and their effectiveness will be a significant determining factor in the success of the method overall. How fun do people consider challenges? Are the prompts and restrictions annoying? Or do they spark creativity? How intrinsically motivating are the challenges? Are users willing to complete challenges without knowing about their security motivations? How many challenges do people complete before becoming bored by the concept? Yet, success of this paradigm may depend on more than just the challenges. What other factors play a role in adoption and continued usage of incremental verification?

The role played by challenge recipients raises a different set of research questions. As discussed in Section 7, these include how to motivate recipients' participation and whether to inform them about the security motivation of their tasks. In addition to answering verification questions, a more implicit responsibility of recipients is to spot suspicious verification videos that may be fake. Subsection 5.4 has argued that, if recipients spot artifacts or inconsistencies, they will discuss this with the sender. This behavior is thus a major part of the defense against potential attacks using deepfakes. However, whether this discussion will actually happen is an open question that future work needs to investigate.

Another research direction could be investigating, or strengthening, the security properties of I3FV. How hard is it for an attacker to forge a challenge video? What additional steps could be added to the protocol to make this more difficult? Are there anti-impersonation safeguards that can be added to the app? Also important to understand is the rate of false positives in the system: how many people will say someone correctly performed the challenge even if they did not? False negatives—in which recipients incorrectly flag legitimate challenge completions as attacks—also need to be investigated. Some may be accidental, while others may represent confusion or misunderstanding. Either way, too-frequent occurrences have the potential to frustrate or alarm users.

The most ecologically valid way of studing I3FV is to implement and deploy it with one of the popular E2EE services. A full-scale deployment is naturally not an easy task; a more feasible alternative is to perform a small-scale study. However, this has its own difficulties. Since the long-term success of the scheme is a concern, it is important to conduct a user study in an environment that resembles a real-world deployment as closely as possible. This necessitates, at a minimum, the following two requirements: First, people must be able to use the app over an extended period of time (i.e., not in a single session). Second, people must engage in interactions with their real friends (i.e., not strangers recruited for the study, or researchers). To meet these requirements, there are two major approaches researchers can take: they could design a custom messaging app that implements the verification scheme, or they could test the verification scheme "on top of" an existing app people use. Each approach has its own advantages and disadvantages.

Designing a custom app has several advantages. Researchers get to control the full experience, which ensures that the challenges come from the app itself. They can also implement and test various gamification features. Overall, this results in a much more natural and realistic user experience. The downside is that this approach suffers in its ecological validity. A research prototype would lack the social networks found on existing apps. If participants are recruited individually, none of their existing contacts will be on the new app, so they will have little motivation to use it. If participants are recruited in groups, most of their contacts will *still* be using other apps, and so there is a significant risk that they revert to their previously-preferred communication channels, especially if the research prototype does not offer as compelling a user experience. These limitations put into question participants' long-term usage of any prototype app—independent of the authentication ceremony implementation—which makes it difficult to accurately answer research questions about people's willingness to use incremental verification over time.

The other option is to perform research with an existing app. Ideally, researchers could modify the client of an E2EE messaging service with additional features, allowing participants to keep using it for their regular communication, but also experience the new verification features researchers are interested in testing. However, due to the tight control most messaging services exert over clients, at present this may not be feasible. Another approach, which may work for studying certain research questions, is not to implement new features, but to ask participants to simulate certain behaviors in their existing apps. For example, participants could receive challenges from researchers over email but then share them with select friends or groups through (for example) Snapchat. This approach allows users to stay on widely used platforms, which can increase long-term engagement. There is even the potential for participants to share their content with a wider circle of friends, not just other study participants, which some may find more motivating. On the negative side, such a study needs to be designed with a way for participants to explain challenge videos to their friends, so as to address questions that may arise and avoid potential awkwardness. This approach also makes it difficult to test the recipient side of the interaction, since recipients cannot verify a video in the app. Additionally, this approach has a much less natural user experience, does not provide a channel for in-app messaging or explanations, and also precludes testing gamification features and other incentives that services may wish to deploy to motivate participation.

Ultimately, researchers should choose the approach that they feel is most helpful for answering their specific research questions, despite the trade-offs it may entail.

Beyond testing the assumptions underpinning the particular version of I3FV proposed in this paper, future work should also consider how its principles could be applied to other authentication ceremonies—either in secure messaging or beyond. In particular, the method that has been the focus of this paper emphasizes emulating the behaviors of apps like Snapchat and TikTok, which have become popular predominantly (though not exclusively) with younger demographics. However, the generalized ideas of incidental incremental in-band fingerprint verification may yield other instantiations. Researchers and designers can experiment to discover what these may be.

This paper has proposed a new paradigm of *incidental incremental in-band fingerprint verification*. It represents a novel method for performing authentication ceremonies and encompasses a design space with many variables. By subjecting the key servers used by E2EE messaging to greater scrutiny, this approach would increase trustworthiness of encrypted communications. However, there may be other approaches for increasing trust, including other authentication ceremonies that are similarly incidental, incremental, and/or in-band; to discover them, more exploration of this paradigm and research problem is needed.

ACKNOWLEDGMENTS

The author was supported by the SPLICE research program through the National Science Foundation (NSF) SaTC Frontiers program under award number 1955805. Thank you to Kent Seamons for shepherding this paper.

REFERENCES

[1] Harold Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael A. Specter, and Daniel J. Weitzner. 2015. Keys under Doormats: Mandating Insecurity by Requiring Government Access to All Data and Communications. *Journal of Cybersecurity* (Nov. 2015), tyv009. https://doi.org/10.1093/cybsec/tyv009

[2] Ruba Abu-Salma, Elissa M. Redmiles, Blase Ur, and Miranda Wei. 2018. Exploring User Mental Models of End-to-End Encrypted Communication Tools. In *8th USENIX Workshop on Free and Open Communications on the Internet (FOCI 18)*. USENIX Association, Baltimore, MD. https://www.usenix.org/conference/foci18/presentation/abu-salma

[3] Ruba Abu-Salma, M. Angela Sasse, Joseph Bonneau, Anastasia Danilova, Alena Naiakshina, and Matthew Smith. 2017. Obstacles to the Adoption of Secure Communication Tools. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Jose, CA, USA, 137–153. https://doi.org/10.1109/SP.2017.65

[4] Noura Alomar, Mansour Alsaleh, and Abdulrahman Alarifi. 2017. Social Authentication Applications, Attacks, Defense Strategies and Future Research Directions: A Systematic Review. *IEEE Communications Surveys & Tutorials* 19, 2 (2017), 1080–1111. https://doi.org/10.1109/COMST.2017.2651741

[5] Martin Anderson. 2022. To Uncover a Deepfake Video Call, Ask the Caller to Turn Sideways. https://metaphysic.ai/to-uncover-a-deepfake-video-call-ask-the-caller-to-turn-sideways/

[6] Wei Bai, Moses Namara, Yichen Qian, Patrick Gage Kelley, Michelle L. Mazurek, and Doowon Kim. 2016. An Inconvenient Trust: User Attitudes toward Security and Usability Tradeoffs for Key-Directory Encryption Systems. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. USENIX Association, Denver, CO, 113–130. https://www.usenix.org/conference/soups2016/technical-sessions/presentation/bai

[7] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. 2006. Fourth-Factor Authentication: Somebody You Know. In *Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06*. ACM Press, Alexandria, Virginia, USA, 168–178. https://doi.org/10.1145/1180405.1180427

[8] Bumble. n.d.. How to Use Bumble's Photo Verification Feature. https://bumble.com/the-buzz/request-verification

[9] Jon Callas. 2019. The 'Ghost User' Ploy to Break Encryption Won't Work. https://www.aclu.org/blog/privacy-technology/ghost-user-ploy-break-encryption-wont-work

[10] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker (Eds.). 2008. *Digital Watermarking and Steganography* (second edition ed.). Morgan Kaufmann, Burlington. https://doi.org/10.1016/B978-012372585-1.50003-6

[11] Nour Dabbour and Anil Somayaji. 2020. Towards In-Band Non-Cryptographic Authentication. In *New Security Paradigms Workshop 2020*. ACM, Online USA, 20–33. https://doi.org/10.1145/3442167.3442180

[12] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. 2016. An Empirical Study of Textual Key-Fingerprint Representations. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 193–208. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/dechand

[13] Facebook. 2013. Introducing Trusted Contacts. https://www.facebook.com/notes/facebook-security/introducing-trusted-contacts/10151362774980766

[14] Matthias Fassl, Lea Theresa Gröber, and Katharina Krombholz. 2021. Exploring User-Centered Security Design for Usable Authentication Ceremonies. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–15. https://doi.org/10.1145/3411764.3445164

[15] Mikhail Fomichev, Flor Alvarez, Daniel Steinmetzer, Paul Gardner-Stephen, and Matthias Hollick. 2018. Survey and Systematization of Secure Device Pairing. *IEEE Communications Surveys & Tutorials* 20, 1 (2018), 517–550. https://doi.org/10.1109/COMST.2017.2748278

[16] M.T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. 2006. Loud and Clear: Human-Verifiable Authentication Based on Audio. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE, Lisboa, Portugal, 10–10. https://doi.org/10.1109/ICDCS.2006.52

[17] Amir Herzberg, Hemi Leibowitz, Kent Seamons, Elham Vaziripour, Justin Wu, and Daniel Zappala. 2021. Secure Messaging Authentication Ceremonies Are Broken. *IEEE Security & Privacy* 19, 2 (March 2021), 29–37. https://doi.org/10.1109/MSEC.2020.3039727

[18] Dayana Hristova, Suzana Jovicic, Barbara Göbl, Sara de Freitas, and Thomas Slunecko. 2022. "Why Did We Lose Our Snapchat Streak?". Social Media Gamification and Metacommunication. *Computers in Human Behavior Reports* 5 (March 2022), 100172. https://doi.org/10.1016/j.chbr.2022.100172

[19] H. C. Hsiao, Y. H. Lin, A. Studer, C. Studer, K. H. Wang, H. Kikuchi, A. Perrig, H. M. Sun, and B. Y. Yang. 2009. A Study of User-Friendly Hash Comparison Schemes. In *2009 Annual Computer Security Applications Conference*. 105–114. https://doi.org/10.1109/ACSAC.2009.20

[20] Ashar Javed, David Bletgen, Florian Kohlar, Markus Durmuth, and Jorg Schwenk. 2014. Secure Fallback Authentication and the Trusted Friend Attack. In *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops*. IEEE, Madrid, Spain, 22–28. https://doi.org/10.1109/ICDCSW.2014.30

[21] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. 2018. Transfer Learning from Speaker Verification to Multispeaker Text-to-Speech Synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 4485–4495.

[22] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. 2016. MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers. *IEEE Transactions on Information Forensics and Security* 11, 6 (June 2016), 1306–1320. https://doi.org/10.1109/TIFS.2015.2505626

[23] Ronald Kainda, Ivan Flechais, and A. W. Roscoe. 2009. Usability and Security of Out-of-Band Channels in Secure Device Pairing Protocols. In *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*. ACM Press, Mountain View, California, 1. https://doi.org/10.1145/1572532.1572547

[24] Keybase. n.d.. Proofs. https://book.keybase.io/account#proofs

[25] Alfred Kobsa, Rahim Sonawalla, Gene Tsudik, Ersin Uzun, and Yang Wang. 2009. Serial Hook-Ups: A Comparative Usability Study of Secure Device Pairing Methods. In *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*. ACM Press, Mountain View, California, 1. https://doi.org/10.1145/1572532.1572546

[26] Arun Kumar, Nitesh Saxena, Gene Tsudik, and Ersin Uzun. 2009. A Comparative Study of Secure Device Pairing Methods. *Pervasive and Mobile Computing* 5, 6 (Dec. 2009), 734–749. https://doi.org/10.1016/j.pmcj.2009.07.008

[27] Philipp Markert, Maximilian Golla, Elizabeth Stobert, and Markus Dürmuth. 2019. Work in Progress: A Comparative Long-Term Study of Fallback Authentication. In *Proceedings 2019 Workshop on Usable Security*. Internet Society, San Diego, CA. https://doi.org/10.14722/usec.2019.23030

[28] R. Mayrhofer and H. Gellersen. 2009. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing* 8, 6 (June 2009), 792–806. https://doi.org/10.1109/TMC.2009.51

[29] J.M. McCune, A. Perrig, and M.K. Reiter. 2005. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. In *2005 IEEE Symposium on Security and Privacy (S&P '05)*. IEEE, Oakland, CA, USA, 110–124. https://doi.org/10.1109/SP.2005.19

[30] Marcela S. Melara, Aaron Blankstein, Joseph Bonneau, Edward W. Felten, and Michael J. Freedman. 2015. CONIKS: Bringing Key Transparency to End Users. In *24th USENIX Security Symposium (USENIX Security 15)*. USENIX Association, Washington, D.C., 383–398. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara

[31] Yisroel Mirsky and Wenke Lee. 2022. The Creation and Detection of Deepfakes: A Survey. *Comput. Surveys* 54, 1 (Jan. 2022), 1–41. https://doi.org/10.1145/3425780

[32] Jack Nicas. 2022. Apple Becomes First Company to Hit $3 Trillion Market Value. *The New York Times* (Jan. 2022). https://www.nytimes.com/2022/01/03/technology/apple-3-trillion-market-value.html

[33] Adrian Perrig and Dawn Song. 1999. Hash Visualization: A New Technique to Improve Real-World Security. In *In International Workshop on Cryptographic Techniques and E-Commerce*. 131–138.

[34] Martin Petraschek, Thomas Hoeher, Oliver Jung, Helmut Hlavacs, and Wilfried Gansterer. 2008. Security and Usability Aspects of Man-in-the-Middle Attacks on ZRTP. *J. UCS* 14 (Jan. 2008), 673–692. https://www.jucs.org/jucs_14_5/security_and_usability_aspects/jucs_14_05_0673_0692_petraschek.pdf

[35] Stuart Schechter, Serge Egelman, and Robert W Reeder. 2009. It's Not What You Know, but Who You Know. (2009), 10.

[36] Maliheh Shirvanian and Nitesh Saxena. 2014. Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Scottsdale Arizona USA, 868–879. https://doi.org/10.1145/2660267.2660274

[37] Jordan Shropshire and Philip Menard. 2015. A New Approach to Mobile Device Authentication. *Proceedings of the Pre-ICIS Workshop on Information Security and Privacy (SIGSEC)* (2015), 17. https://aisel.aisnet.org/wisp2015/22

[38] Signal. 2022. Story Time. https://signal.org/blog/introducing-stories/

[39] Snapchat. n.d.. Create Community Filters. https://support.snapchat.com/en-US/article/user-submitted-geofilters

[40] Snapchat. n.d.. Filter Ads. https://forbusiness.snapchat.com/advertising/ad-formats/filters

[41] Ahren Studer, Timothy Passaro, and Lujo Bauer. 2011. Don't Bump, Shake on It: The Exploitation of a Popular Accelerometer-Based Smart Phone Exchange and Its Secure Replacement. In *Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC '11*. ACM Press, Orlando, Florida, 333. https://doi.org/10.1145/2076732.2076780

[42] Joshua Tan, Lujo Bauer, Joseph Bonneau, Lorrie Faith Cranor, Jeremy Thomas, and Blase Ur. 2017. Can Unicorns Help Users Compare Crypto Key Fingerprints?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3787–3798. https://doi.org/10.1145/3025453.3025733

[43] Alexander Ulrich, Ralph Holz, Peter Hauck, and Georg Carle. 2011. Investigating the OpenPGP Web of Trust. In *Computer Security – ESORICS 2011*, Vijay Atluri and Claudia Diaz (Eds.). Vol. 6879. Springer Berlin Heidelberg, Berlin, Heidelberg, 489–507. https://doi.org/10.1007/978-3-642-23822-2_27

[44] Nik Unger, Sergej Dechand, Joseph Bonneau, Sascha Fahl, Henning Perl, Ian Goldberg, and Matthew Smith. 2015. SoK: Secure Messaging. In *2015 IEEE Symposium on Security and Privacy*. IEEE, San Jose, CA, USA, 232–249. https://doi.org/10.1109/SP.2015.22

[45] Serge Vaudenay. 2005. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005. Proceedings*, Victor Shoup (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 309–326. https://doi.org/10.1007/11535218_19

[46] Elham Vaziripour, Devon Howard, Jake Tyler, Mark O'Neill, Justin Wu, Kent Seamons, and Daniel Zappala. 2019. I Don't Even Have to Bother Them!: Using Social Media to Automate the Authentication Ceremony in Secure Messaging. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300323

[47] Elham Vaziripour, Justin Wu, Mark O'Neill, Ray Clinton, Jordan Whitehead, Scott Heidbrink, Kent Seamons, and Daniel Zappala. 2017. Is That You, Alice? A Usability Study of the Authentication Ceremony of Secure Messaging Applications. In *Symposium on Usable Privacy and Security (SOUPS)*.

[48] Elham Vaziripour, Justin Wu, Mark O'Neill, Daniel Metro, Josh Cockrell, Timothy Moffett, Jordan Whitehead, Nick Bonner, Kent Seamons, and Daniel Zappala. 2018. Action Needed! Helping Users Find and Complete the Authentication Ceremony in Signal. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association, Baltimore, MD, 47–62. https://www.usenix.org/conference/soups2018/presentation/vaziripour

[49] Noel Warford, Tara Matthews, Kaitlyn Yang, Omer Akgul, Sunny Consolvo, Patrick Gage Kelley, Nathan Malkin, Michelle L. Mazurek, Manya Sleeper, and Kurt Thomas. 2022. SoK: A Framework for Unifying At-Risk User Research. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 2344–2360. https://doi.org/10.1109/SP46214.2022.9833643

[50] Anna Wiener. 2020. Taking Back Our Privacy. *The New Yorker* (Oct. 2020). https://www.newyorker.com/magazine/2020/10/26/taking-back-our-privacy

[51] Justin Wu, Cyrus Gattrell, Devon Howard, Jake Tyler, Elham Vaziripour, Daniel Zappala, and Kent Seamons. 2019. "Something Isn't Secure, but I'm Not Sure How That Translates into a Problem": Promoting Autonomy by Designing for Understanding in Signal. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. USENIX Association, Santa Clara, CA, 137–153. https://www.usenix.org/conference/soups2019/presentation/wu

[52] Justin Wu and Daniel Zappala. 2018. When Is a Tree Really a Truck? Exploring Mental Models of Encryption. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. USENIX Association, Baltimore, MD, 395–409. https://www.usenix.org/conference/soups2018/presentation/wu

[53] Tarun Kumar Yadav, Devashish Gosain, Amir Herzberg, Daniel Zappala, and Kent Seamons. 2022. Automatic Detection of Fake Key Attacks in Secure Messaging. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Los Angeles CA USA, 3019–3032. https://doi.org/10.1145/3548606.3560588

[54] P Zimmermann, A. Johnston, and J Callas. 2011. ZRTP: Media Path Key Agreement for Unicast Secure RTP. https://www.ietf.org/rfc/rfc6189.txt