The Security of Lazy Users in Out-of-Band Authentication

Moni Naor[1(✉)], Lior Rotem[2], and Gil Segev[2]

[1] Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, 76100 Rehovot, Israel
moni.naor@weizmann.ac.il

[2] School of Computer Science and Engineering, Hebrew University of Jerusalem,
91904 Jerusalem, Israel
{lior.rotem,segev}@cs.huji.ac.il

Abstract. Faced with the threats posed by man-in-the-middle attacks, messaging platforms rely on "out-of-band" authentication, assuming that users have access to an external channel for authenticating one short value. For example, assuming that users recognizing each other's voice can authenticate a short value, Telegram and WhatApp ask their users to compare 288-bit and 200-bit values, respectively. The existing protocols, however, do not take into account the plausible behavior of users who may be "lazy" and only compare parts of these values (rather than their entirety).

Motivated by such a security-critical user behavior, we study the security of lazy users in out-of-band authentication. We start by showing that both the protocol implemented by WhatsApp and the statistically-optimal protocol of Naor, Segev and Smith (CRYPTO '06) are completely vulnerable to man-in-the-middle attacks when the users consider only a half of the out-of-band authenticated value. In this light, we put forward a framework that captures the behavior and security of lazy users. Our notions of security consider both statistical security and computational security, and for each flavor we derive a lower bound on the tradeoff between the number of positions that are considered by the lazy users and the adversary's forgery probability.

Within our framework we then provide two authentication protocols. First, in the statistical setting, we present a transformation that converts any out-of-band authentication protocol into one that is secure even when executed by lazy users. Instantiating our transformation with a new refinement of the protocol of Naor et al. results in a protocol whose tradeoff essentially matches our lower bound in the statistical setting.

M. Naor—Incumbent of the Judith Kleeman Professorial Chair. Supported in part by a grant from the Israel Science Foundation.

L. Rotem and G. Segev—Supported by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253), by the Israel Science Foundation (Grant No. 483/13), by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11), and by the US-Israel Binational Science Foundation (Grant No. 2014632).

A. Beimel and S. Dziembowski (Eds.): TCC 2018, LNCS 11240, pp. 575–599, 2018.
https://doi.org/10.1007/978-3-030-03810-6_21

Then, in the computational setting, we show that the computationally-optimal protocol of Vaudenay (CRYPTO '05) is secure even when executed by lazy users – and its tradeoff matches our lower bound in the computational setting.

1 Introduction

Instant messaging platforms are gaining increased popularity and hold an overall user base of more than 1.5 billion active users (e.g., WhatsApp, Signal, Telegram and many more [Wik]). These platforms recognize user authentication and end-to-end encryption as key ingredients for ensuring secure communication within them, and extensive efforts are currently put into the security of messaging, both commercially (e.g., [PM16,Telb,Wha,Vib]) and academically (e.g., [FMB+16, BSJ+17,CCD+17,KBB17]). A key challenge in securing messaging platforms is that of protecting against man-in-the-middle attacks when setting up secure end-to-end channels. This is exacerbated by the ad-hoc nature of these platforms.

Out-of-Band Authentication. Faced with the threats posed by man-in-the-middle attacks, existing messaging platforms enable "out-of-band" authentication, assuming that users have access to an *external* channel for authenticating short values. These values are typically derived from the public keys of the users, or more generally from the transcript of any key-exchange protocol that the users execute for setting up a secure end-to-end channel.

For example, some messaging platforms offer users the ability to compare with each other a value that is displayed by their devices (see Telegram [Tela], WhatsApp [Wha], Viber [Vib] and more [Mem17]). This relies on the assumption two users can establish a *low-bandwidth authenticated channel* (e.g., by recognizing each other's voice): A man-on-the-middle adversary can view, delay or even remove any message sent over this channel, but cannot undetectably modify its content.

Such an authentication model that assumes a low-bandwidth authenticated channel was considered back in 1984 by Rivest and Shamir [RS84].[1] More recently, this model was formalized by Vaudenay [Vau05] in the computational setting (i.e., considering computationally-bounded adversaries) and extended by Naor et al. [NSS06,NSS08] to the statistical setting (i.e., considering computationally-unbounded adversaries) and by Rotem and Segev [RS18] to the group setting. The out-of-band message authentication problem considers a sender that would like to authenticate a message m to a receiver.[2] The

[1] Rivest and Shamir proposed the "Interlock" protocol which enables two users, who recognize each other's voice, to mutually authenticate their public keys in the absence of a trusted infrastructure. Potential attacks on the Interlock protocol were identified later on [BM94,Ell96].

[2] As mentioned above, for messaging platforms the message m typically corresponds to the public keys of the users or to the transcript of any key-exchange protocol that they execute.

users communicate over two channels: An insecure channel over which a man-in-the-middle adversary has complete control, and a low-bandwidth authenticated channel, enabling the sender to "out-of-band" authenticate one short value. The security requirement asks for an upper bound on any man-in-the-middle adversary's probability of fooling the receiver into accepting a fraudulent message.

An Effort vs. Security Tradeoff. Given that the out-of-band channel has only low bandwidth, research on out-of-band authentication has so far focused on constructing protocols that offer the best-possible tradeoff between the length of their out-of-band authenticated values (corresponding to the amount of effort required from the users) and their security (corresponding to the adversary's forgery probability). Vaudenay [Vau05], Naor et al. [NSS06] and Rotem and Segev [RS18] provided complete characterizations of this tradeoff in their above-mentioned respective settings, providing both lower bounds and protocols that match them. However, these protocols rely on the assumption that the *human users* indeed follow the protocol in its entirety. In particular, they rely on the assumption that the users out-of-band authenticate the *entire* value that the protocols instruct them to authenticate.

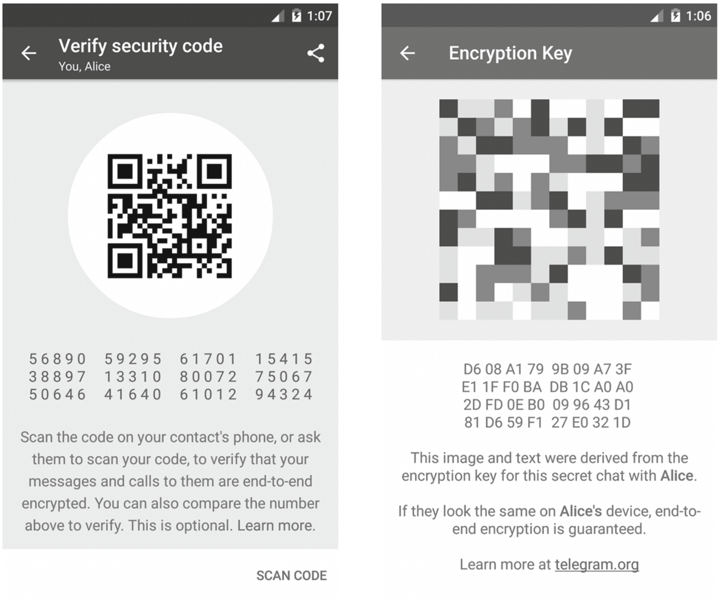
This assumption, however, may not always be realistic: The lengths of the out-of-band authenticated values offered by the existing messaging platforms may not align with the potential effort of different users. Specifically, existing messaging platforms ask their users to out-of-band authenticate values whose lengths range from roughly 200 bits (e.g., WhatsApp and Signal) to 288 bits (e.g., Telegram) – see Fig. 1. Given that the out-of-band channel in implemented in these platforms via a manual comparison operation, the security of such protocols must take into account users that may compare only a subset of the positions of these values. We refer to such users, who out-of-band authenticate only a substring of the protocol's out-of-band authenticated value, as "lazy users".

As repeatedly demonstrated by research on usable security and human-computer interaction, it is rather likely that a substantial part of the messaging platforms' user base may in fact be considered lazy (see, for example, [LS03,PLF03,BA04,Her09,HZF+14,AFJ15,DDB+16] and the references therein). This state of affairs, where a security-critical user behavior is not taken into account, is extremely bothering.

1.1 Our Contributions

Motivated by the above-described plausible and security-critical behavior of "lazy" users, we put forward a framework that captures the behavior and security of such users in out-of-band authentication. Within our framework we characterize the possible security guarantees for lazy users by presenting protocols together with essentially matching lower bounds both in the computational setting and in the statistical setting. Our main contributions are as follows.

The Insecurity of Existing Protocols. We strengthen our motivation by showing that the protocol implemented by WhatsApp [Wha] and the protocol of Naor et al. [NSS06] are completely vulnerable to man-in-the-middle attacks



56890 59295 61701 15415
38897 13310 80072 75067
50646 41640 61012 94324

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

D6 08 A1 79 9B 09 A7 3F
E1 1F F0 BA DB 1C A0 A0
2D FD 0E B0 09 96 43 D1
81 D6 59 F1 27 E0 32 1D

This image and text were derived from the encryption key for this secret chat with **Alice**.

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

Learn more at telegram.org

Fig. 1. Out-of-band authentication in WhatsApp and Telegram. WhatApp and Telegram (as well as many other messaging platforms) implement the out-of-band channel by asking their users to manually compare two strings. WhatApp (on the right) asks its users to manually compare 60 decimal digits corresponding to an out-of-band authenticated value [Wha] of about 200 bits. Telegram (on the left) asks its users to manually compare 64 characters corresponding to a 288-bit out-of-band authenticated value [Telc]. The images are taken from [Mem17].

when the parties consider only a half (or fewer) of the characters of the out-of-band authenticated value. This demonstrates that it is not only the case that the existing protocols do not take security-critical user behavior into account, they may in fact become completely insecure when executed by lazy users. In the following section, we discuss the main underlying reason for these protocols' vulnerability, and how our constructions overcome it.

Modeling the Behavior and Security of Lazy Users. We put forward a framework that captures the behavior and security of lazy users. Our notions of security consider both computational security and statistical security, and for each flavor we derive a lower bound on the tradeoff between the number of positions that are considered by the lazy users out of the out-of-band authenticated value and the adversary's forgery probability. These lower bounds are summarized in Table 1, and we refer the reader to Sect. 1.3 for a more detailed overview.

Immunizing Statistically-Secure Protocols Against Lazy Users. Recall that the statistically-secure protocol of Naor et al. [NSS06] becomes completely insecure when executed by lazy users. Intuitively, this is the case because the

Table 1. Summary of our results – protocols vs. lower bounds. We denote by $\mathcal{I}$ the subset of positions of the out-of-band authenticated value that the users consider, by Σ the alphabet over which the out-of-band authenticated value is defined, and by n the length of the sender's input message. Our computationally-secure protocol relies on the existence of any one-way function (see Theorem 6.1), whereas our statistically-secure protocol and our two lower bounds do not rely on any computational assumptions (see Corollary 5.2, Theorem 7.1 and Corollary 7.3). Note that our upper bound and lower bound in the computational setting match within an additive 2^{-n} term (which is a significantly lower-order term for not-too-short input messages). In the statistical setting our bounds match within a constant factor (in addition to the additive 2^{-n} term).

	Our protocols		Our lower bounds
	Forgery probability	Alphabet size	
Computational security	$2^{-\lvert\mathcal{I}\rvert}$	2	$2^{-\lvert\mathcal{I}\rvert\cdot\log\lvert\Sigma\rvert} - 2^{-n}$
Statistical security	$2^{-\lvert\mathcal{I}\rvert}$	2^8	$2^{-\lvert\mathcal{I}\rvert\cdot\log\lvert\Sigma\rvert/2} - 2^{-n}$

influence of each bit of the sender's input message (i.e., the message to be authenticated) is not "well-spread" across the out-of-band authenticated value (see Sect. 4 for an in-depth discussion).

Addressing this property, we provide a transformation that converts any statistically-secure protocol (that does not necessarily provide any security for lazy users) into a protocol that is statistically-secure for lazy users. Instantiating our transformation with the protocol of Naor et al. results in a concrete statistically-secure protocol for lazy users. Moreover, in the full version of the paper [NRS18] we show that by refining the protocol of Naor et al. the resulting instantiation uses an alphabet whose size is as small as 2^8 – which nearly matches our above-mentioned lower bound in the statistical setting.[3] We stress that our transformation and the protocol resulted from applying it to the protocol of Naor et al. are oblivious to the subset $\mathcal{I}$ of positions that users eventually read or even to the number of positions they read. Meaning, we provide a *single* protocol that guarantees security for every possible subset $\mathcal{I}$. An interesting open question is whether a protocol which is statistically-secure for lazy users can be constructed over a binary alphabet.

In fact, our transformation can also be applied to any computationally-secure protocol that satisfies a natural parallel composability guarantee. However, as shown by our next result, this is somewhat unnecessary.

Matching the Optimal Tradeoff for Computationally-Secure Protocols. Whereas the statistically-optimal protocol of Naor et al. is completely insecure for lazy users, we show that the computationally-optimal protocol of

[3] As we discuss in more detail in Sect. 1.3, when moving to the setting of lazy users, the size of the alphabet over which the out-of-band authenticated value is defined becomes of great importance. This is in contrast to the traditional (non-lazy) setting, in which this has no impact on security.

Vaudenay [Vau05] is optimally secure for lazy users as well. Intuitively, this is due to the following observation: Even though the out-of-band authenticated value in this protocol is determined independently of the sender's input message (which is reminiscent of the protocol of Naor et al. in the statistical setting), the protocol "ties together" the message and the out-of-band authenticated value *in their entirety* using a non-malleable commitment scheme (which, in practice, can be replaced by a hash function modeled as a random oracle). Note that as in the statistical setting, the protocol is oblivious to the particular subset of positions that the users eventually consider.

Extensions. We also discuss possible extensions of our framework. First, in the full version [NRS18], we consider the notion of *adaptive laziness*, which gives the adversary the ability to choose the subset of positions to be considered by the users even *after* the out-of-band authenticated value is determined. Although we find this notion somewhat less motivated in the context of lazy users, we nevertheless extend our definitions and proofs of security to this stronger notion.

Second, we note that our notions of security, lower bounds and protocols naturally extend to the group setting considered by Rotem and Segev [RS18]. Specifically, in the computational setting the protocol of Rotem and Segev can be shown to be optimally-secure for lazy users; and in the statistical setting, our general transformation can be easily adapted to support group protocols (and can then be instantiated with the statistically-secure protocol of Rotem and Segev).

1.2 Related Work

Bounds for Out-of-Band Authentication. In the standard setting of out-of-band authentication (i.e., with non-lazy users), Vaudenay [Vau05] and Vaudenay and Pasini [PV06] established tight bounds for the tradeoff between the length of the (entire) out-of-band authenticated value and the adversary's forgery probability in the computational setting. They provided a protocol [Vau05] in which the forgery probability is bounded by $2^{-\ell}$, where ℓ is the bit-length of the out-of-band authenticated value, and a matching lower bound [PV06]. Naor et al. [NSS06] observed a gap between the computational and the statistical settings: They proved that the forgery probability in the statistical setting of any protocol is always at least $2^{-\ell/2}$, and provided a protocol that matches this lower bound within a constant factor. We refer the reader to Table 2 for a summary of these bounds, and note that our results provide a similar characterization for lazy users in both the computational and the statistical settings (recall Table 1).

The Security of Messaging Platforms. Many recent works addressed the goals of formalizing the security guarantees of messaging platforms, as well as analyzing the security of the protocols used by these platforms and identifying potential weaknesses within them – see, for example, [FMB+16, HL16, BSJ+17, CCD+17, CGCG+17, CGC17, KBB17, SKH17, RMS18, Gre18a, Gre18b] and the references therein. Throughout this extensive line of research, the security of

Table 2. Previous work – protocols vs. lower bounds. We denote by ℓ the length of the out-of-band authenticated value and by n the length of the sender's input message. The computationally-secure protocol of Vaudenay [Vau05] relies on the existence of any one-way function, whereas the statistically-secure protocol of Naor et al. [NSS06] and the two lower bounds [NSS06, PV06] do not rely on any computational assumptions.

	Protocols	Lower bounds
Computational security [Vau05, PV06]	$2^{-\ell}$	$2^{-\ell} - 2^{-n}$
Statistical security [NSS06]	$O\left(2^{-\ell/2}\right)$	$2^{-\ell/2} - 2^{-n}$

messaging protocols assumes an initial authentication phase for avoiding man-in-the-middle attacks. As mentioned in most of the afore-listed references, such an initial authentication phase is based on out-of-band authentication.

1.3 Overview of Our Contributions

We extend the existing framework for out-of-band authentication protocols [Vau05, PV06, NSS06, RS18] to accommodate the security-critical behavior of "lazy users", that may consider only a certain part of the out-of-band authenticated value (e.g., its left-most half, its right-most 10 characters, or a few randomly-chosen positions). We model this behavior by having the sender send only a substring of the out-of-band authenticated value, and requiring that for any such substring the man-in-the-middle attacker's forgery probability is bounded by some pre-defined parameter associated with it. That is, whereas a standard (i.e., "non-lazy") out-of-band authentication protocol is parameterized by an upper bound $\epsilon \in (0,1)$ on the adversary's forgery probability, a protocol in our framework is parameterized by a function $\epsilon(\cdot)$ which maps every subset $\mathcal{I}$ of positions of the out-of-band authenticated value to an associated upper bound $\epsilon(\mathcal{I})$.[4]

In addition, our definitions also extend those of Vaudenay and Naor et al. by accounting for out-of-band authentication values over *non-binary* alphabets (indeed, in the existing real-world implementations of out-of-band authentication protocols, the out-of-band authenticated value is displayed to the users as a string over some non-binary alphabet – recall Fig. 1). When the users are assumed to consider the entire out-of-band authenticated value, the particular choice of alphabet (and alphabet size) is mainly a matter of providing a convenient user interface. In the presence of lazy users, however, the size of the alphabet of the out-of-band authenticated value plays an important role in what may be referred to as the "granularity" of the users' laziness.

[4] Note that protocols in our framework must explicitly address (in terms of both completeness and soundness) the case where only part of the out-of-band authenticated value is considered. This is the case, in particular, in our motivating example where verification is done by comparing the out-of-band authenticated string to a value that is computed by the receiver.

Let us consider for concreteness a pair of users that read some 32 bits out of a 64-bit out-of-band authenticated value. If the out-of-band authenticated value is simply a 64-bit string (i.e., over a binary alphabet), then the users may possibly read any of the $\binom{64}{32} > 1.83 \times 10^{18}$ many 32-bit substrings of it. On the other hand, if the alphabet is of larger size, say 8 characters, the users' ability to partially access the out-of-band authenticated value is more coarse-grained. In particular, they can still read only a substring of the authenticated value, but are restricted to reading specific blocks of consecutive 8 bits in their entirety. In other words, users that read 32 bits in this setting may read only one of $\binom{8}{4} = 70$ many 32-bit substrings of the out-of-band authenticated value.

Identifying the Weakness in Existing Protocols. It is quite simple to construct a contrived example of a secure protocol that is completely insecure when executed by lazy users. Thus, we chose to focus on the protocols of WhatsApp [Wha] and Naor et al. [NSS06] for the following reasons: (1) the protocol implemented by WhatsApp is among the most widely-used out-of-band authentication protocols, and (2) the protocol of Naor et al. offers the optimal tradeoff between the length of the out-of-band authenticated value and the adversary's forgery probability in the statistical setting (thus showing that both computationally-secure protocols and statistically-secure ones may become completely insecure when executed by lazy users).

Analyzing our rather simple attacks on these protocols (see Sect. 4), we identify a key property that they have in common which makes them completely insecure when executed by lazy users: Intuitively, different sections of the sender input message (i.e., the message m to be authenticated) influence different sections of the out-of-band authenticated value. Hence, if the users only consider a subset of positions of the out-of-band authenticated value that is independent in some sense from a particular part of the message to be authenticated, the adversary can replace this part of the message in an undetected manner (we refer to this property as "over locality"). In what follows, we discuss why the protocol of Vaudenay in the computational setting does not suffer from over locality; and how our general transformation in the statistical setting addresses it.

Naive Approaches that Fail. A potential approach to immunizing any comparison based out-of-band authentication protocol against lazy users, is to have the parties run the protocol and then hash the out-of-band authenticated value with a random oracle (in addition to transmitting it over the insecure channel). On the face of it, this resolves any over dependency on locality the initial protocol might have exhibited. However, this approach may generally suffer from the major shortcoming of introducing a tradeoff between the adversary's running time and its success probability (aside, of course, from relying on a random oracle which may be undesirable if the security of the underlying protocol does not require it). More concretely, an adversary that runs in time $T(\lambda)$ has forgery probability that is roughly (at least) $T(\lambda)/2^{-|\mathcal{I}|}$, where $\mathcal{I}$ is the subset of positions that the parties consider. When $\mathcal{I}$ is small (which is exactly the case with lazy users), then the asymptotics "do not kick in", and the latter forgery probability is significant. This is precisely the reason why we are interested in protocols in which for every such subset $\mathcal{I}$, the forgery probability is bounded by $\epsilon(\mathcal{I}) + \nu(\lambda)$ (where $\nu(\cdot)$ is a negligible function of the security parameter λ) *for every* polynomial-time adversary.

An additional potential approach is to have the parties apply some fixed error-correcting code to the out-of-band authenticated value. Though this may have the effect of increasing the fraction of inconsistent positions in the out-of-band authenticated value at the end of any forgery attempt, it does not provide the security guarantees we seek: If before applying the error-correcting code there was some subset of t positions for some fixed t, for which there was an attack causing the receiver to output a fraudulent message with probability ϵ, this may still be the case after applying the code. Moreover, this approach has the consequence of worsening the tradeoff between the length of the out-of-band authenticated value and the adversary's forgery probability. Similarly, adding redundancy to the input message itself (e.g., by applying an error-correcting code to it) is not necessarily helpful in immunizing protocols against lazy users.

Another possibility is to reduce the number of characters in the out-of-band authenticated value by mapping it to a larger alphabet. As discussed above, this has the effect of restricting the lazy behavior of the users; in particular, assuming that the users read at least one character of the out-of-band value, after increasing the alphabet size, this single character constitutes a larger fraction of the out-of-band value. Alas, even if the new alphabet is sufficiently large so that the out-of-band value consists just of two characters, the resulting protocol may still be insecure for lazy users who read only one of them (this is the case, for example, with the protocols of WhatsApp [Wha] and Naor et al. [NSS06]). On the other hand, our lower bounds on the bit-length of the out-of-band value (see Sect. 7) imply that in order for the out-of-band value to consists only of a single character, its alphabet size has to be at least $1/\epsilon$, where ϵ is the forgery probability. For any reasonable level of security, this means an impractical-sized alphabet has to be used.

Security for Lazy Users via "Influence Spreading". Our transformation in the statistical setting takes as input a parameter $t \in \mathbb{N}$ and any statistically-secure out-of-band authentication π with out-of-band authenticated value of length ℓ and forgery probability at most ϵ. It proceeds by having the sender S and the receiver R run t parallel executions of π with the same input message m to S. Afterwards, S parses each of the resulting t out-of-band authentication values as a single character from an alphabet of the appropriate size, concatenates them into a single string of length t (over the larger alphabet) and sends it over the out-of-band channel. When considering some subset $\mathcal{I} \subseteq [t]$ of the characters in the new out-of-band authenticated value, the receiver R accepts the message m if and only if it accepts m in each of the executions corresponding to the subset $\mathcal{I}$. We show that for every subset $\mathcal{I} \subseteq [t]$, the forgery probability in this new protocol is bounded by $\epsilon'(\mathcal{I}) \leq \epsilon^{|\mathcal{I}|}$.

In light of our observations regarding protocols that are insecure for lazy users, this transformation can be thought of in the following manner: We start with a protocol that might be insecure for lazy users and suffer from over locality, and we "spread" the influence of each bit of the input message across all characters of the new out-of-band authenticated value via the parallel invocations of the basic protocol.

When instantiated with the protocol of Naor et al. [NSS06] (while setting its security to $\epsilon = 1/2$), our transformation yields a protocol with a constant-size alphabet which is statistically-secure for lazy users: For every subset $\mathcal{I} \subseteq [t]$, the forgery probability corresponding to $\mathcal{I}$ is bounded by $2^{-|\mathcal{I}|}$. However, using the protocol of Naor et al. and their analysis "off the shelf" results in an alphabet which is, though constant-size, large and impractical (concretely, it is of size $2^{16} = 65536$). Hence, in the full version [NRS18], we show by a refined analysis of the protocol of Naor et al. that this constant can be reduced to $2^8 = 256$ (which fits nicely, for example, in the set of 333 emoji Telegram uses as the alphabet in the verification of their voice calls).

Leveraging the "Local Sensitivity" of Non-malleable Commitments. Informally speaking, the protocol of Vaudenay [Vau05] consists of the following steps: (1) On input m, S sends m to R, chooses a random r_S and commits to the message (m, r_S); (2) R sends a random r_R to S; (3) S reveals r_S; and (4) S sends $r_S \oplus r_R$ over the out-of-band authenticated channel. In the lazy user setting, where the users only read the subset $\mathcal{I}$ of positions in the out-of-band authenticated value, R accepts m if and only if the value $(r_S \oplus r_R)_{\mathcal{I}}$ sent over the out-of-band channel is consistent with her view of the protocol.

In Sect. 6 we prove that when the commitment scheme used in Step (1) is a non-malleable commitment scheme, then this protocol is optimal for lazy users (considering the matching lower bound from Sect. 7). Our proof goes about by considering all potential synchronizations that a man-in-the middle attacker might impose while attacking an execution of the protocol, and showing that in each of them, an attack on the protocol that succeeds with probability noticeably larger than $2^{-|\mathcal{I}|}$ can be translated into an attack on a different property of the underlying commitment scheme.

From a more conceptual point of view, our proof leverages the fact that the non-malleability of commitment schemes is a property which is "locally sensitive" in the following sense. Informally, in a non-malleable commitment scheme, it should be impossible, given a commitment c to some value v, to produce a related commitment $\widehat{c}$ for some value $\widehat{v}$ such that v and $\widehat{v}$ satisfy *any* efficiently recognizable relation. This includes, in particular, relations that are defined with respect to a subset of the positions in v and $\widehat{v}$; and namely, the relation induced by a successful forgery in Vaudenay's protocol when the users only consider the subset $\mathcal{I}$ of positions of the out-of-band authenticated value.

1.4 Paper Organization

The remainder of this paper is organized as follows. In Sect. 2 we present the notation and basic definitions that are used in this work. In Sect. 3 we introduce our framework for modeling the behavior and security of lazy users in out-of-band message authentication protocols. In Sect. 4 we show that existing out-of-band authentication protocols may become completely insecure when executed by lazy users. In Sects. 5 and 6 we present statistically-secure and computationally-secure out-of-band authentication protocols, respectively. Finally, in Sect. 7 we derive lower bounds on the tradeoff between the adversary's forgery probability and the length of the out-of-band authenticated value in out-of-band authentication protocols that are executed by lazy users.

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. For a string s and a subset $\mathcal{I} \subseteq [|s|]$ of positions, we let $s_{\mathcal{I}}$ (sometimes we may write $(s)_{\mathcal{I}}$) denote the substring of s obtained by concatenating the characters of s in the positions specified by the set $\mathcal{I}$ in increasing order. A function $\nu : \mathbb{N} \rightarrow \mathbb{R}^+$ is *negligible* if for any polynomial $p(\cdot)$ there exists an integer N such that for all $n > N$ it holds that $\nu(n) \leq 1/p(n)$.

Shannon Entropy. For a random variable X defined over a finite domain Ω, we rely the standard notion of Shannon entropy: $\mathrm{H}(X) = -\sum_{x \in \Omega} \Pr[X = x] \cdot \log_2 \Pr[X = x]$. Note that for any such X it holds that $\mathrm{H}(X) \leq \log_2 |\Omega|$.

Non-malleable Commitment Schemes [DDN00]**.** We rely on the notion of statistically-binding non-malleable commitments (for basic definitions and background on commitment schemes, we refer the reader to [Gol01]). We follow the indistinguishability-based definition of Lin and Pass [LP11], though we find it convenient to consider non-malleability with respect to content, other than with respect to identities. Intuitively speaking, a non-malleable commitment scheme has the following guarantee: Any efficient adversary cannot use a commitment to some value v in order to produce a commitment to a value $\widehat{v}$ which is "non-trivially" related to v. For formal definitions regarding commitment schemes and non-malleable commitment schemes in particular, see the full version [NRS18].

Dolev et al. [DDN00] constructed non-malleable commitment schemes from any one-way function. Subsequently, Lin and Pass [LP11] and Goyal [Goy11] have shown that constant-round non-malleable commitments can be constructed from the same assumption. The round complexity was further improved by Goyal et al. [GRR+14] to 4 rounds, and by Goyal et al. [GPR16] to 3 rounds assuming the existence of an injective one-way function. Such schemes can also be constructed efficiently in a simple manner in the random-oracle model [BR93]. For further information regarding non-malleable commitment schemes in the standard model see the references above as well as, for example, [Bar02, PR08, LP09, PPV08, PW10, Wee10, GLO+12] and the references therein.

3 Modeling the Security of Lazy Users

In this section we introduce our framework for modeling the behavior and security of lazy users in out-of-band message authentication protocols. We start by reviewing the communication model and existing notions of security for out-of-band message authentication [Vau05, NSS06], and then present our notions of security for the case of lazy users.

3.1 Out-of-Band Authentication

Following the framework of Vaudenay [Vau05] and Naor et al. [NSS06], we model the interaction between the sender and the receiver as occurring over two types of channels: A bidirectional insecure channel that is completely vulnerable to man-in-the middle attacks, and an authenticated unidirectional low-bandwidth channel from the sender to the receiver. The adversary is assumed to have complete control over the insecure channel: She can read, delay and remove any messages sent by the two parties, as well as insert new messages of her choice at any point in time. In particular, this provides the adversary with considerable control over the synchronization of the protocol's execution. Nonetheless, the execution is still guaranteed to be "marginally synchronized": Each party sends her message in the ith round of the protocol only upon receiving the due message of round $i-1$. As for the out-of-band channel, we assume that the sender is equipped with a low-bandwidth channel, through which the sender may send a short message to the receiver in an authenticated manner (but without any secrecy guarantee). The adversary may read or remove this message, and may delay it for different periods of time, but cannot modify it in an undetectable manner.

We follow the definitions of Vaudenay [Vau05] and Naor et al. [NSS06], generalizing naturally to consider out-of-band authenticated values over general alphabets and not only over the binary alphabet. As we discuss later on, this is of little importance in the standard setting (where the parties are assumed to read the entire out-of-band authenticated value), but will play a significant role when considering lazy users. Following Naor et al. we differentiate between protocols that are computationally secure and ones that are statistically secure. We formalize the notion of *statistically-secure* out-of-band authentication protocols as:

Definition 3.1. *Let $n, \ell, r \in \mathbb{N}$, let $\epsilon \in (0,1)$ and let Σ be an alphabet. A statistically-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol over Σ is an r-round protocol in which the sender S is invoked on an n-bit message and sends at most ℓ characters of Σ over the out-of-band authenticated channel. The following requirements must hold:*

1. **Correctness:** *In an honest execution of the protocol, for any input message $m \in \{0,1\}^n$ on which S is invoked, R outputs m with probability* 1.

2. **Unforgeability:** *For any man-in-the-middle adversary A and for any adversarially chosen input message $m \in \{0,1\}^n$ on which S is invoked, the probability that R outputs some message $\widehat{m} \notin \{m, \perp\}$ in an execution with S that is attacked by A is at most ϵ.*

A *computationally-secure* out-of-band authentication protocol is defined similarly, except that security need only hold against efficient adversaries, and the probability of forgery is also allowed to additively grow (with respect to the statistical setting) by a negligible function of the security parameter.

Definition 3.2. *Let $n = n(\lambda), \ell = \ell(\lambda), r = r(\lambda), \epsilon = \epsilon(\lambda)$, and $\Sigma = \Sigma(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. A computationally-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol over alphabet Σ is an r-round protocol in which the sender S is invoked on an n-bit message and sends at most ℓ characters of Σ over the out-of-band authenticated channel. The following requirements must hold:*

1. **Correctness:** *In an honest execution of the protocol, for any input message $m \in \{0,1\}^n$ on which S is invoked, R outputs m with probability 1.*
2. **Unforgeability:** *For any probabilistic polynomial-time man-in-the-middle adversary A there exists a negligible function $\nu(\cdot)$ such that: For any input message $m \in \{0,1\}^n$ chosen by the adversary and on which S is invoked, the probability that R outputs some message $\widehat{m} \notin \{m, \perp\}$ in an execution with S that is attacked by A is at most $\epsilon + \nu(\lambda)$.*

3.2 The Security of Lazy Users

In order to formally capture the lazy-users setting, given an out-of-band authentication protocol we define a collection of "lazy protocols", one per each possible subset of positions of the out-of-band authenticated value. Informally speaking, given a protocol π in which the out-of-band authenticated value consists of ℓ characters, for a subset $\mathcal{I} \subseteq [\ell]$ of indexes, we consider the "lazy protocol" $\pi_{\mathcal{I}}$ in which the parties execute π, with the exception that S only sends over the out-of-band channel the substring of the out-of-band authenticated value that corresponds to the positions in the set $\mathcal{I}$.

Specifically, let π be a (statistically-secure or computationally-secure) out-of-band (n, ℓ, r, ϵ)-authentication protocol over an alphabet Σ (recall Definitions 3.1 and 3.2). For every subset $\mathcal{I} \subseteq [\ell]$ of the positions of its out-of-band authenticated value, the "lazy protocol" $\pi_{\mathcal{I}}$ is defined as follows:

1. On input $m \in \{0,1\}^n$ to S, the sender S and receiver R run the first $r-1$ rounds of π. Let $v \in \Sigma^\ell$ be the out-of-band authenticated value that S is due to send in round r.
2. S receives $\mathcal{I}$ and sends only $v_{\mathcal{I}}$ over the out-of-band authenticated channel.
3. R receives $\mathcal{I}$ and $v_{\mathcal{I}}$, and decides on her output according to π.[5]

[5] As noted before, the protocols we consider in this paper must be defined for every substring of the out-of-band authenticated value.

Using this notion, Definitions 3.3 and 3.4 below formalize the extensions discussed above in the statistical setting and computational setting, respectively. Intuitively, we define the security of out-of-band authentication protocols for lazy users by letting the bound on the forgery probability be a function of the subset $\mathcal{I}$ considered by the users. Concretely, an out-of-band authentication protocol π is parameterized by some function ϵ, which maps each possible set of positions $\mathcal{I}$ of the out-of-band authenticated value to be read by the users to a matching upper bound on the forgery probability. That is, in case the users only read the out-of-band authentication value in positions $\mathcal{I}$, an adversary should be able to make the receiver output a fraudulent message with probability at most $\epsilon(\mathcal{I})$. This approach has the benefit of being very general on the one hand, while coinciding with the standard definitions (see Definitions 3.1 and 3.2) when $\mathcal{I} = [\ell]$. We note, however, that one may still consider a more restrictive notion where the forgery probability should only depend on the size of $\mathcal{I}$ (observe that this is a strict restriction of our notion).

Definition 3.3. *Let* $n, \ell, r \in \mathbb{N}$ *and let* $\epsilon : 2^{[\ell]} \to [0, 1]$. *A protocol* π *is* a statistically-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol for lazy users *over alphabet* Σ *if for every* $\mathcal{I} \subseteq [\ell]$ *the protocol* $\pi_{\mathcal{I}}$ *is a statistically-secure out-of-band* $(n, |\mathcal{I}|, r, \epsilon(\mathcal{I}))$*-authentication protocol.*

Definition 3.4. *Let* $n = n(\lambda), \ell = \ell(\lambda), r = r(\lambda)$ *and* $\Sigma = \Sigma(\lambda)$ *be functions of the security parameter* $\lambda \in \mathbb{N}$, *and let* $\epsilon = \epsilon(\lambda, \cdot) : 2^{[\ell]} \to [0, 1]$. *A protocol* π *is* a computationally-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol for lazy users *over alphabet* Σ *if for every* $\mathcal{I} = \mathcal{I}(\lambda) \subseteq [\ell]$ *the protocol* $\pi_{\mathcal{I}}$ *is a computationally-secure out-of-band* $(n, |\mathcal{I}|, r, \epsilon(\cdot, \mathcal{I}))$*-authentication protocol.*

4 The Insecurity of Existing Protocols

In this section we show that existing out-of-band authentication protocols may become completely insecure when executed by lazy users. We focus on the computationally-secure protocol implemented by WhatsApp [Wha] and on the statistically-secure protocol of Naor et al. [NSS06], and show that these protocols are completely vulnerable to man-in-the-middle attacks when the parties consider only a half (or less) of the out-of-band authenticated value.

Concretely, for each of these two protocols we present an efficient man-in-the-middle attacker that fools the receiver into accepting a fraudulent message with probability 1. Then, we discuss the basic underlying structure that these two protocols share, which makes them completely insecure when executed by lazy users.

WhatsApp's Protocol [Wha]**.** Consider any protocol where in order to authenticate a message m, the sender S partitions m into two halves $m = m_1 \| m_2$, and authenticates each half using some out-of-band authentication protocol separately and independently. The out-of-band authenticated value is then $\sigma = \sigma_1 \| \sigma_2$, where σ_1 and σ_2 are the out-of-band authenticated values of the two executions. If the underlying out-of-band authentication protocol is secure and the users read the entire string σ, then this newly-defined protocol is secure as well (though, possibly, with a sub-optimal tradeoff between the adversary's forgery probability and the length of the out-of-band authenticated value). However, consider for example the case where the parties only read σ_1 (or a substring of it). In this case, no security is guaranteed and a man-in-the-middle adversary can trivially make R output a fraudulent message of the form $\widehat{m} = m_1 \| \widehat{m_2}$ for some $\widehat{m_2} \neq m_2$. A similar problem arises when the parties read only σ_2 (or a substring of it).

The above protocol might seem like a pathological example, specifically contrived for our needs, but this is in fact exactly the approach used by WhatsApp. Concretely, a pair of WhatsApp users wishing to verify that each of them has the correct key of the other user compare a 60-digit sequence displayed on each of their screens. This sequence is derived by hashing each user's key into a 30-digit string, and concatenating the two strings.[6] It is not hard to see that if the users only compare the first half of the out-of-band authenticated value, it might very well be the case that one of them holds a fraudulent key, completely compromising the secrecy of their chat.

The Protocol of Naor et al. [NSS06]**.** Naor et al. [NSS06] presented a construction of a statistically-secure out-of-band authentication protocol that relies on the following idea. Loosely speaking, the two parties iteratively hash the message into shorter intermediate values until reaching a short enough value that can be transmitted out-of-band. More concretely, in each round of the protocol the parties cooperatively choose an algebraic hash function: They treat the input message and the intermediate values as polynomials over finite fields of appropriate sizes, and in each round, one party chooses a random element in the field on which the polynomial is evaluated, and the other party chooses a random shift to apply to the result. When choosing the last hash function, the sender S is the one to choose the element on which the polynomial is evaluated. The out-of-band authenticated value then consists of two parts: (1) The result of the last hash function (according to the view of S); (2) and the last element S chose.

Yet again, if the parties read and compare the entire out-of-band authenticated value, then Naor et al. proved that this protocol is secure (and provides the optimal tradeoff between the adversary's forgery probability and the length of the out-of-band authenticated value). Alas, if the users are lazy, and read only one of the two parts of the out-of-band authenticated value, then the protocol becomes completely insecure. Concretely, if the parties only read the part that corresponds to the last field element chosen by S, then a trivial attack exists:

[6] From WhatsApp's security white paper [Wha, p. 10]: "WhatsApp users additionally have the option to verify the keys of the other users with whom they are communicating so that they are able to confirm that an unauthorized third party (or WhatsApp) has not initiated a man-in-the-middle attack. This can be done by scanning a QR code, or by comparing a 60-digit number. [...] The 60-digit number is computed by concatenating the two 30-digit numeric fingerprints for each user's Identity Key".

The man-in-the-middle adversary simply runs two independent executions, one with the sender S and one with the receiver R, on two different input messages, with the exception of choosing the same field element as S does in the last hash function of her interaction with R.

Summary: The Underlying Weakness. The property that both of the above examples share and which makes them completely insecure in the face of rather trivial attacks can be articulated in the following manner: In both cases, different sections of the input message to be authenticated affect different sections of the out-of-band authenticated value. In the case of WhatsApp, each user's key affects only half of the out-of-band authentication value (but both keys should be verified). In the case of Naor et al. [NSS06], the input message to be authenticated goes into the computation of only half of the out-of-band authenticated value, while the other half is simply a random value generated during the execution of the protocol.

It is instructive to view our positive results also in this light, as this may provide the reader with additional intuition regarding the security of our constructions:

1. In the statistical setting, our transformation (and its resulting protocol when instantiated with that of Naor et al. [NSS06]) can be interpreted as follows. We start with an out-of-band authentication protocol that guarantees no security for lazy users to begin with (but does guarantee security for users who fully comply with the protocol), and in particular may suffer from the same problematic property described above. We transform this protocol into a protocol that provides security for lazy users by "spreading" the influence of each bit of the input message m across all characters of the out-of-band authenticated value of the resulting protocol.
2. In the computational setting we consider Vaudenay's protocol [Vau05] whose out-of-band authenticated value is simply a uniformly-distributed string that is generated during the execution of the protocol. Intuitively speaking, even though this value is determined independently of the input message, we "tie together" the message *in its entirety* and the out-of-band authenticated value using cryptographic tools (namely, a non-malleable commitment scheme).

5 Immunizing Statistically-Secure Protocols Against Lazy Users

In this section we present a generic transformation that uses any out-of-band authentication protocol that is secure under a certain form of parallel repetition for constructing an out-of-band authentication protocol for lazy users. In particular, our transformation can be applied to any statistically-secure protocol, and can thus be instantiated with the protocol of Naor et al. [NSS06]. As our transformation itself is statistically secure, this yields a statistically-secure protocol (that comes very close to matching our lower bound on the tradeoff between adversary's forgery probability and the length of the partial out-of-band authenticated value considered by the lazy users – see Corollary 7.3).

We first present and analyze our transformation for statistically-secure protocols, as well as discuss the properties of its instantiation with the protocol of Naor et al. [NSS06]. Then, we discuss the specific composability property required of computationally-secure protocols in order for them to be compatible with our transformation (this, however, is somewhat less motivated given that our computationally-secure protocol in Sect. 6 already matches our lower bound in the computational setting).

The Transformation. The building block underlying our transformation is an out-of-band authentication protocol that does not necessarily guarantee any form of security for lazy users. Loosely speaking, our transformation proceeds as follows: On input message m, the parties run ℓ parallel and independent executions of the underlying protocol with the same message m, and parse each of the resulting ℓ out-of-band authentication values as a single character from an alphabet of the appropriate size. The sender S then concatenates these ℓ characters into a single string of length ℓ (over the larger alphabet) and sends it over the out-of-band authenticated channel. In a lazy execution of the protocol, where the receiver considers only some number $t \leq \ell$ out of the ℓ out-of-band authenticated characters, the receiver accepts m if and only if it m is accepted in each of the corresponding t executions.

Intuitively, if the forgery probability of the underlying protocol is bounded by ϵ', then fooling a receiver that reads only a predetermined t-character subset of the out-of-band authenticated value requires the adversary to break the unforgeability (in the standard sense, not considering lazy users) of t copies of the underlying protocol, and hence the adversary's forgery probability is bounded by $(\epsilon')^t$ in the statistical setting.

More formally, let $n', \ell', r' \in \mathbb{N}$, let $\epsilon' \in (0, 1)$, and let π' is a statistically-secure out-of-band $(n', \ell', r', \epsilon')$-authentication protocol; that is, π' is an r'-round protocol for out-of-band authentication of messages of length n', where the sender out-of-band authenticates at most ℓ' bits, and the probability of forgery is bounded by ϵ'. We use π' to construct a statistically-secure out-of-band $(n = n', \ell, r = r', \epsilon)$-authentication protocol for lazy users, denoted π_{Lazy}, for any $\ell \in \mathbb{N}$, such that $\epsilon(\mathcal{I}) = (\epsilon')^{|\mathcal{I}|}$ for every $\mathcal{I} \subseteq [\ell]$.

The protocol for lazy users, denoted π_{Lazy}, is defined as follows for every $\mathcal{I} \subseteq [\ell]$ (i.e., this is the "lazy protocol" $\pi_{\mathsf{Lazy},\mathcal{I}}$ – see Sect. 3):

1. On input message m to S, S and R run ℓ parallel executions of π' up to (and including) round $r' - 1$ with the same input message m to S in all executions. Denote the out-of-band authenticated values that S computes in these executions by $\sigma_1 \cdots \sigma_\ell \in \{0, 1\}^{\ell'}$.
2. For each $i \in [\ell]$, S parses σ_i as a single character over an alphabet of size $k = 2^{\ell'}$; denote the ith character by β_i. S then receives $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\} \subseteq [\ell]$ and sends $\sigma = \beta_{i_1} \| \ldots \| \beta_{i_{|\mathcal{I}|}}$ over the out-of-band authenticated channel.
3. R receives $\mathcal{I}$, parses $\sigma = \sigma_{i_1} \cdots \sigma_{i_{|\mathcal{I}|}}$ as $|\mathcal{I}|$ binary strings of length ℓ' each. For every $i \in \mathcal{I}$, denote by $\widehat{m_i}$ the output of R in the ith execution given

R's view of that execution (including σ_i). If for every $i, j \in \mathcal{I}$ it holds that $\widehat{m_i} = \widehat{m_j}$, then R outputs $\widehat{m_{i_1}}$. Otherwise, R outputs $\perp$.

The correctness and security of the protocol π_{Lazy} are stated in the following theorem.

Theorem 5.1. *Let π' be a statistically-secure out-of-band (n, ℓ', r, ϵ') -authentication protocol, let $k = 2^{\ell'}$ and let $\ell \in \mathbb{N}$. Then, π_{Lazy} is a statistically-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol for lazy users over an alphabet of size k, where $\epsilon(\mathcal{I}) = (\epsilon')^{|\mathcal{I}|}$ for every $\mathcal{I} \subseteq [\ell]$.*

The correctness and round complexity of π_{Lazy} follow immediately from the correctness and round complexity of π', respectively. The unforgeability of π_{Lazy} for lazy users (vis-à-vis Definition 3.3) is proven in the full version [NRS18], yielding the above theorem.

A Concrete Instantiation. Naor et al. [NSS06] constructed a statistically-secure out-of-bound (n, ℓ', r, ϵ')-authentication protocol for any $n, r \in \mathbb{N}$ and any $\epsilon' \in (0, 1)$, where $\ell' \leq \log(1/\epsilon') + \log^{(r-1)} + O(1)$. Instantiating our protocol π_{Lazy} with the protocol of Naor et al. as π', while setting $r = \Omega(\log^* n)$ and $\epsilon' = 1/2$, yields a statistically-secure out-of-band authentication protocol for lazy users with the same round complexity and a constant-size alphabet. This is formalized by the following corollary.

Corollary 5.2. *For any $n, \ell \in \mathbb{N}$, there exists a statistically-secure out-of-band $(n, \ell, \log^* n, \epsilon)$-authentication protocol for lazy users over a constant size alphabet, where $\epsilon(\mathcal{I}) = 2^{-|\mathcal{I}|}$ for every $\mathcal{I} \subseteq [\ell]$.*

In the full version [NRS18] we also provide a refined analysis of the protocol of Naor et al. which reduces the alphabet size of the protocol from Corollary 5.2 to 2^8, and discuss how our transformation applies to computationally-secure protocols with some specific parallel-composability property.

6 Matching the Optimal Tradeoff for Computationally-Secure Protocols

In this section we show that Vaudenay's computationally-secure protocol [Vau05] can be extended to allow execution by lazy users, and that the resulting protocol matches our lower bound on the tradeoff between the adversary's forgery probability and the length of the out-of-band authenticated value for lazy users (see Theorem 7.1). That is, the protocol offers the optimal tradeoff between the adversary's forgery probability and the length of the partial out-of-band authenticated value considered by the lazy users.

The basic building block used by the protocol is any non-malleable statistically-binding commitment scheme Com. From a foundational point of view, such a scheme with a constant number of rounds can be constructed based on any one-way function in the standard model, and from a more practical point of view, such a scheme can be constructed by simply invoking a hash function modeled as a random oracle (see Sect. 2).

The protocol, which we denote by π_{Comp}, is parametrized by the security parameter $\lambda \in \mathbb{N}$, the message length $n = n(\lambda) \in \mathbb{N}$ and the length of the out-of-band authenticated value $\ell = \ell(\lambda) \in \mathbb{N}$, and is defined as follows:

1. On input the security parameter $\lambda \in \mathbb{N}$ and a message $m \in \{0,1\}^n$, the sender S chooses a random $r_S \leftarrow \{0,1\}^\ell$, sends m to the receiver R, and commits to the pair (m, r_S) to receiver R using Com. Denote the resulting commitment by c_S and its corresponding decommitment by d_S.[7] Denote the message and commitment as received by R by $\widehat{m}$ and $\widehat{c_S}$, respectively.
2. The receiver R chooses a random $r_R \leftarrow \{0,1\}^\ell$ and sends it to the sender S. Denote by $\widehat{r_R}$ the value that S receives.
3. The sender S sends the decommitment d_S to R. Denote by $\widehat{d_S}$ the decommitment R receives. If $\widehat{d_S}$ is not a valid decommitment to $\widehat{c_S}$ or if the revealed value is not of the form $(\widehat{m}, *)$, then R outputs $\perp$. Otherwise, let $(\widehat{m}, \widehat{r_S})$ be the revealed value.
4. The sender S sends $\sigma = r_S \oplus \widehat{r_R}$ over the out-of-band channel. R checks if $\widehat{r_S} \oplus r_R = \sigma$. If so, R outputs $\widehat{m}$, and otherwise R outputs $\perp$.

The following theorem captures the security of the above protocol, stating that it provides the optimal tradeoff as discussed above.

Theorem 6.1. *Let $n = n(\cdot), r = r(\cdot)$ and $\ell = \ell(\cdot)$ be functions of the security parameter $\lambda \in \mathbb{N}$ and let Com be an r-round statistically-binding non-malleable commitment scheme. Then, protocol π_{Comp} is a computationally-secure out-of-band $(n, \ell, r+3, \epsilon)$-authentication protocol for lazy users (over the alphabet $\Sigma = \{0,1\}$), where $\epsilon(\lambda, \mathcal{I}) = 2^{-|\mathcal{I}|}$ for every $\lambda \in \mathbb{N}$ and for every $\mathcal{I} \subseteq [\ell(\lambda)]$.*

Our protocol incurs an almost minimal overhead in the number of rounds relative to the round complexity of the underlying commitment scheme: The number of rounds of insecure communication is $r+2$ (this includes the $r+1$ rounds necessary for commitment and decommitment), to which we add only a single message over the insecure channel, and a single message over the out-of-band authenticated channel. In the plain model, a non-malleable commitment is known to exist with $r = 3$, while in the random oracle model, there exist non-interactive non-malleable commitments (i.e., with $r = 1$).

The security proof of our protocol considers all possible synchronizations a man-in-the-middle adversary may impose on an execution of the protocol. For each such synchronization and for every possible subset $\mathcal{I} \subseteq [\ell]$ of positions of the out-of-band authenticated value, we bound the forgery probability by $2^{-|\mathcal{I}|}+$

[7] As a commitment scheme may be interactive, when referring to a commitment, we mean the transcript of the interaction between the committer and the receiver during an execution of the commit phase of the commitment scheme. When the scheme is non-interactive, a commitment is simply a single string sent from the committer to the receiver.

$\nu(\lambda)$, for a negligible function $\nu(\lambda)$, by converting an adversary achieving better forgery probability into an adversary that breaks a specific security property of the underlying commitment scheme (i.e., binding, hiding or non-malleability). The full proof is given in the full version [NRS18].

7 Lower Bounds on the Security of Lazy Users

Vaudenay [Vau05] and Naor et al. [NSS06] established tight bounds on the trade-off between the adversary's forgery probability and the length of the out-of-band authenticated value in out-of-band authentication. In this section we show that their lower bounds, in both the computational and statistical setting, directly translate into corresponding lower bounds for protocols that are executed by lazy users.

7.1 Computationally-Secure Protocols

In any computationally-secure out-of-band authentication protocol where the probability of forgery is bounded by $\epsilon > 0$, the sender must out-of-band authenticate at least $\log(1/\epsilon)$ bits. This can be seen, for example, by analyzing the collision probability of the random variable corresponding to the out-of-band authenticated value (see for example, [PV06]). Below, we show that this reasoning generalizes to the case of lazy users: Namely, for each number $k \in [\ell]$ of bits read from the out-of-band authenticated value, we provide a corresponding lower bound.

Theorem 7.1. *For any computationally-secure out-of-band (n, ℓ, r, ϵ) -authentication protocol for lazy users over alphabet Σ, it holds that*

$$\epsilon(\mathcal{I}) \geq 2^{-|\mathcal{I}|\cdot\log|\Sigma|} - 2^{-n}$$

for every $\mathcal{I} \subseteq [\ell]$.

Proof. Let π be any computationally-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol for lazy users over alphabet Σ. Let $\lambda \in \mathbb{N}$ and $\ell = \ell(\lambda)$ and fix any $\mathcal{I} \subseteq [\ell]$. Consider the following attack:

1. Choose a random $m \leftarrow \{0,1\}^n$ and run an honest execution with S on input m (with the adversary playing the role of R). Denote by v the out-of-band authenticated value S sends at the end of the execution. Delay the relaying of v to (the real) R until the end of the attack.
2. Choose a random $\widehat{m} \leftarrow \{0,1\}^n$ and run an honest execution with R, where the adversary plays the role S on input $\widehat{m}$. Denote by $\widehat{v}$ the out-of-band authenticated value that the simulated sender sends at the end of the execution. If $\widehat{v}_{\mathcal{I}} = v_{\mathcal{I}}$, forward v to R; otherwise, terminate.

Denote by $V_{\mathcal{I}}$ the random variable corresponding to the substring of the out-of-band authenticated value defined by the positions in $\mathcal{I}$, where the distribution of $V_{\mathcal{I}}$ is induced by an honest execution of π on a randomly chosen input message to S. Then, the following holds:

$$\begin{aligned}\Pr_{(\widehat{v}_{\mathcal{I}},v_{\mathcal{I}})\leftarrow V_{\mathcal{I}}\times V_{\mathcal{I}}}\left[\widehat{v}_{\mathcal{I}} = v_{\mathcal{I}}\right] &= \sum_{v_{\mathcal{I}}} \left(\Pr\left[V_{\mathcal{I}} = v_{\mathcal{I}}\right]\right)^2 = 2^{\log \sum_{v_{\mathcal{I}}}(\Pr[V_{\mathcal{I}}=v_{\mathcal{I}}])^2} \\ &\geq 2^{\sum_{v_{\mathcal{I}}} \Pr[V_{\mathcal{I}}=v_{\mathcal{I}}]\log(\Pr[V_{\mathcal{I}}=v_{\mathcal{I}}])} = 2^{-\mathrm{H}(V_{\mathcal{I}})}.\end{aligned}$$

The inequality above follows from Jensen's inequality.

Let $\mathsf{Forge}_{\mathcal{I}}$ denote the event in which the above attack goes through; i.e., R outputs a fraudulent message. By the correctness of π, it holds that

$$\begin{aligned}\Pr\left[\mathsf{Forge}_{\mathcal{I}}\right] &\geq \Pr\left[\widehat{v}_{\mathcal{I}} = v_{\mathcal{I}} \wedge \widehat{m} \neq m\right] \\ &\geq \Pr\left[\widehat{v}_{\mathcal{I}} = v_{\mathcal{I}}\right] - \Pr\left[\widehat{m} = m\right] \\ &\geq 2^{-\mathrm{H}(V_{\mathcal{I}})} - 2^{n}.\end{aligned}$$

On the one hand, by the unforgeability of π, it must hold that $\epsilon(\mathcal{I}) \geq 2^{-\mathrm{H}(V_{\mathcal{I}})} - 2^{n}$. On the other hand, it is always the case that $\mathrm{H}(V_{\mathcal{I}}) \leq |\mathcal{I}| \cdot \log|\Sigma|$. Taken together, these inequalities yield the theorem. ■

The lower bound of Theorem 7.1 should be thought of in the following terms. On the one hand, if the message to be authenticated is short (relative to the bandwidth of the out-of-band authenticated channel), then the sender can just go ahead and send it over the out-of-band channel. On the other hand, if it is long, then the term 2^{-n} is small and of little significance, and the attack from our proof succeeds with probability close to $2^{-|\mathcal{I}|\cdot\log|\Sigma|}$. Specifically, for any protocol in which the length of the out-of-band authenticated value is independent of the length of the input message to be authenticated, the success probability of our attack can be made arbitrarily close to $2^{-|\mathcal{I}|\cdot\log|\Sigma|}$ (while considering arbitrarily long input messages).

7.2 Statistically-Secure Protocols

Naor et al. [NSS06] proved a lower bound on the length of the out-of-band authenticated value in any statistically-secure out-of-band authentication protocol. More precisely, they provided a lower bound on the Shannon entropy of the random variable corresponding to the out-of-band authenticated value. If we denote this random value by V, the lower bound of Naor et al. can be articulated as follow:

Theorem 7.2 ([NSS06]). *For any statistically-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol it holds that*

$$\epsilon \geq 2^{-\mathrm{H}(V)/2} - 2^{-n}$$

Theorem 7.2 implies the following, more general, lower bound for out-of-band authentication protocols for lazy users over possibly non-binary alphabets.

Corollary 7.3. *For any statistically-secure out-of-band (n, ℓ, r, ϵ)-authentication protocol for lazy users over alphabet Σ, it holds that for every $\mathcal{I} \subseteq [\ell]$*

$$\epsilon(|\mathcal{I}|) \geq 2^{-|\mathcal{I}|\cdot\log(|\Sigma|)/2} - 2^{-n}.$$

Proof. Let π be any (n, ℓ, r, ϵ)-authentication protocol for lazy users over alphabet Σ. By definition, this means that for any $\mathcal{I} \subseteq [\ell]$, the induced protocol $\pi_{\mathcal{I}}$ is an $(n, |\mathcal{I}|, r, \epsilon(\mathcal{I}))$-authentication protocol. For every $\mathcal{I} \subseteq [\ell]$, denote by $V_{\mathcal{I}}$ the random variable corresponding to the substring of the out-of-band authenticated value that is induced by the subset $\mathcal{I}$. Hence, by Theorem 7.2, for every $\mathcal{I} \subseteq [\ell]$ it holds that

$$\epsilon(|\mathcal{I}|) \geq 2^{-\mathrm{H}(V_{\mathcal{I}})/2} - 2^{-n}.$$

For every $\mathcal{I} \subseteq [\ell]$ it holds that $\mathrm{H}(V_{\mathcal{I}}) \leq |\mathcal{I}| \cdot \log|\Sigma|$, and combining this fact with the above inequality completes the proof. ■

References

[AFJ15] Alghamdi, D., Flechais, I., Jirotka, M.: Security practices for households bank customers in the kingdom of Saudi Arabia. In: Symposium on Usable Privacy and Security (SOUPS), pp. 297–308 (2015)

[BA04] Besnard, D., Arief, B.: Computer security impaired by legitimate users. Comput. Secur. **23**(3), 253–264 (2004)

[Bar02] Barak, B.: Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 345–355 (2002)

[BM94] Bellovin, S.M., Merritt, M.: An attack on the interlock protocol when used for authentication. IEEE Trans. Inf. Theor. **40**(1), 273–275 (1994)

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73 (1993)

[BSJ+17] Bellare, M., Singh, A.C., Jaeger, J., Nyayapati, M., Stepanovs, I.: Ratcheted encryption and key exchange: the security of messaging. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 619–650. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_21

[CCD+17] Cohn-Gordon, K., Cremers, C.J.F., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the signal messaging protocol. In: Proceedings of the 2nd IEEE European Symposium on Security and Privacy (EuroS&P), pp. 451–466 (2017)

[CGC17] Cohn-Gordon, K., Cremers, C.: Mind the gap: where provable security and real-world messaging don't quite meet. Cryptology ePrint Archive, Report 2017/982 (2017)

[CGCG+17] Cohn-Gordon, K., Cremers, C., Garratt, L., Millican, J., Milner, K.: On ends-to-ends encryption: asynchronous group messaging with strong security guarantees. Cryptology ePrint Archive, Report 2017/666 (2017)

[DDB+16] Dupree, J.L., Devries, R., Berry, D.M., Lank, E.: Privacy personas: clustering users via attitudes and behaviors toward security practices. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 5228–5239. ACM (2016)

[DDN00] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. SIAM J. Comput. **30**(2), 391–437 (2000)

[Ell96] Ellison, C.M.: Establishing identity without certification authorities. In: Proceedings of the 6th USENIX Security Symposium, p. 7 (1996)

[FMB+16] Frosch, T., Mainka, C., Bader, C., Bergsma, F., Schwenk, J., Holz, T.: How secure is TextSecure? In: Proceedings of the 1st IEEE European Symposium on Security and Privacy (EuroS&P), pp. 457–472 (2016)

[GLO+12] Goyal, V., Lee, C.-K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: a black-box approach. In: Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, pp. 51–60 (2012)

[Gol01] Goldreich, O.: Foundations of Cryptography: Basic Techniques, vol. 1. Cambridge University Press, Cambridge (2001)

[Goy11] Goyal, V.: Constant round non-malleable protocols using one way functions. In: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, pp. 695–704 (2011)

[GPR16] Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Proceedings of the 48th annual ACM Symposium on Theory of Computing, pp. 1128–1141 (2016)

[Gre18a] Green, M.: Attack of the week: Group messaging in WhatsApp and Signal. A Few Thoughts on Cryptographic Engineering (2018). https://blog.cryptographyengineering.com/2018/01/10/attack-of-the-week-group-messaging

[Gre18b] Greenberg, A.: WhatsApp security flaws could allow snoops to slide into group chats. Wired Magazine (2018). https://www.wired.com/story/whatsapp-security-flaws-encryption-group-chats

[GRR+14] Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science, pp. 41–50 (2014)

[Her09] Herley, C.: So long and no thanks for the externalities: the rational rejection of security advice by users. In: Proceedings of the Workshop on New Security Paradigms, pp. 133–144 (2009)

[HL16] Herzberg, A., Leibowitz, H.: Can Johnny finally encrypt?: evaluating E2E-encryption in popular IM applications. In: Proceedings of the 6th Workshop on Socio-Technical Aspects in Security and Trust, pp. 17–28 (2016)

[HZF+14] Harbach, M., Zezschwitz, E.V., Fichtner, A., Luca, A.D., Smith, M.: It's a hard lock life: a field study of smartphone (un)locking behavior and risk perception. In: Symposium on Usable Privacy and Security (SOUPS), pp. 213–230 (2014)

[KBB17] Kobeissi, N., Bhargavan, K., Blanchet, B.: Automated verification for secure messaging protocols and their implementations: a symbolic and computational approach. In: Proceedings of the 2nd IEEE European Symposium on Security and Privacy (EuroS&P), pp. 435–450 (2017)

[LP09] Lin, H., Pass, R.: Non-malleability amplification. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 189–198 (2009)

[LP11] Lin, H., Pass, R.: Constant-round non-malleable commitments from any one-way function. In: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, pp. 705–714 (2011)

[LS03] Li, S., Shum, H.-Y.: Secure human-computer identification against peeping attacks (SecHCI): A survey (2003)

[Mem17] Membe, T.: A look at how private messengers handle key changes. Medium (2017). https://medium.com/@pepelephew/a-look-at-how-private-messengers-handle-key-changes-5fd4334b809a

[NRS18] Naor, M., Rotem, L., Segev, G.: The security of lazy users in out-of-band authentication. Cryptology ePrint Archive, Report 2018/823 (2018)

[NSS06] Naor, M., Segev, G., Smith, A.: Tight bounds for unconditional authentication protocols in the manual channel and shared key models. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 214–231. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_13

[NSS08] Naor, M., Segev, G., Smith, A.D.: Tight bounds for unconditional authentication protocols in the manual channel and shared key models. IEEE Trans. Inf. Theor. **54**(6), 2408–2425 (2008)

[PLF03] Patrick, A.S., Long, A.C., Flinn, S.: HCI and security systems. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 1056–1057 (2003)

[PM16] Perrin, T., Marlinspike, M.: The double ratchet algorithm (2016). https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf. Accessed 16 May 2018

[PPV08] Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive one-way functions and applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 57–74. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_4

[PR08] Pass, R., Rosen, A.: New and improved constructions of nonmalleable cryptographic protocols. SIAM J. Comput. **38**(2), 702–752 (2008)

[PV06] Pasini, S., Vaudenay, S.: An optimal non-interactive message authentication protocol. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 280–294. Springer, Heidelberg (2006). https://doi.org/10.1007/11605805_18

[PW10] Pass, R., Wee, H.: Constant-round non-malleable commitments from subexponential one-way functions. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 638–655. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_32

[RMS18] Rösler, P., Mainka, C., Schwenk, J.: More is less: on the end-to-end security of group chats in signal, WhatsApp, and Threema. In: Proceedings of the 3nd IEEE European Symposium on Security and Privacy (EuroS&P) (2018)

[RS84] Rivest, R.L., Shamir, A.: How to expose an eavesdropper. Commun. ACM **27**(4), 393–395 (1984)

[RS18] Rotem, L., Segev, G.: Out-of-band authentication in group messaging: computational, statistical, optimal. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 63–89. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_3

[SKH17] Schliep, M., Kariniemi, I., Hopper, N.: Is Bob sending mixed signals? In: Proceedings of the 2017 on Workshop on Privacy in the Electronic Society, pp. 31–40 (2017)

[Tela] Telegram. End-to-end encrypted voice calls - key verification. https://core.telegram.org/api/end-to-end/voice-calls#key-verification. Accessed 16 May 2018

[Telb] Telegram. End-to-end encryption. https://core.telegram.org/api/end-to-end. Accessed 16 May 2018

[Telc] Telegram. FAQ for the technically inclined - hash collisions for Diffie-Hellman keys. https://core.telegram.org/techfaq#hash-collisions-for-diffie-hellman-keys. Accessed 16 May 2018

[Vau05] Vaudenay, S.: Secure communications over insecure channels based on short authenticated strings. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 309–326. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_19

[Vib] Viber encryption overview. https://www.viber.com/app/uploads/Viber-Encryption-Overview.pdf. Accessed 16 May 2018

[Wee10] Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, pp. 531–540 (2010)

[Wha] WhatsApp encryption overview. https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf. Accessed 16 May 2018

[Wik] Wikipedia. Instant messaging. https://en.wikipedia.org/wiki/Instant_messaging. Accessed 16 May 2018