Out-of-Band Authentication in Group Messaging: Computational, Statistical, Optimal

Lior Rotem[*] Gil Segev[†]

Abstract

Extensive efforts are currently put into securing messaging platforms, where a key challenge is that of protecting against man-in-the-middle attacks when setting up secure end-to-end channels. The vast majority of these efforts, however, have so far focused on securing *user-to-user* messaging, and recent attacks indicate that the security of *group* messaging is still quite fragile.

We initiate the study of out-of-band authentication in the group setting, extending the user-to-user setting where messaging platforms (e.g., Telegram and WhatsApp) protect against man-in-the-middle attacks by assuming that users have access to an external channel for authenticating one short value (e.g., two users who recognize each other's voice can compare a short value). Inspired by the frameworks of Vaudenay (CRYPTO '05) and Naor et al. (CRYPTO '06) in the user-to-user setting, we assume that users communicate over a completely-insecure channel, and that a group administrator can out-of-band authenticate one short message to all users. An adversary may read, remove, or delay this message (for all or for some of the users), but cannot undetectably modify it.

Within our framework we establish tight bounds on the tradeoff between the adversary's success probability and the length of the out-of-band authenticated message (which is a crucial bottleneck given that the out-of-band channel is of low bandwidth). We consider both computationally-secure and statistically-secure protocols, and for each flavor of security we construct an authentication protocol and prove a lower bound showing that our protocol achieves essentially the best possible tradeoff.

In particular, considering groups that consist of an administrator and k additional users, for statistically-secure protocols we show that at least $(k+1)\cdot(\log(1/\epsilon)-\Theta(1))$ bits must be out-of-band authenticated, whereas for computationally-secure ones $\log(1/\epsilon)+\log k$ bits suffice, where ϵ is the adversary's success probability. Moreover, instantiating our computationally-secure protocol in the random-oracle model yields an efficient and practically-relevant protocol (which, alternatively, can also be based on any one-way function in the standard model).

A preliminary version of this work appeared in *Advances in Cryptology – CRYPTO '18*.

[*] School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem 91904, Israel. Email: {lior.rotem,segev}@cs.huji.ac.il. Supported by the Israel Science Foundation (Grant No. 483/13) and by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11).

Contents

1 Introduction

Instant messaging is gaining extremely-increased popularity as a tool enabling users to communicate with other users either individually or within groups. A variety of available messaging platforms hold an overall user base of more than 1.5 billion active users (e.g., WhatsApp, Signal, Telegram, and many more [Wik]), and recognize user authentication and end-to-end encryption as key ingredients for ensuring secure communication within them.

Extensive efforts are currently put into securing messaging platforms, both commercially (e.g., [PM16, Telb, Wha]) and academically (e.g., [FMB$^+$16, BSJ$^+$17, CCD$^+$17, KBB17]). The vast majority of these efforts, however, have so far focused on securing *user-to-user* messaging, and substantially less attention has been devoted to securing *group* messaging. Unfortunately, it recently turned out that whereas the security of user-to-user messaging is gradually reaching a stable ground, the security of group messaging is still quite fragile [CGCG$^+$17, RMS18, Gre18a, Gre18b].

Out-of-band authentication. 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. Such attacks are enabled by the inability of users to authenticate their incoming messages given the somewhat ad-hoc nature of messaging platforms.[1] To this end, various messaging platforms enable "out-of-band" authentication, assuming that users have access to an *external* channel for authenticating short values. These values typically correspond to short hash values that are derived, for example, 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, in the user-to-user setting, some messaging platforms offer users the ability to compare with each other a value that is displayed by their devices (e.g., Telegram [Tela], WhatsApp [Wha] and Viber [Vib]).[2] This may rely on the realistic assumption that by recognizing each other's voice, two users can establish a *low-bandwidth authenticated channel*: A man-on-the-middle adversary can view, delay or even remove any message sent over this channel, but cannot modify its content in an undetectable manner.

Such an authentication model was initially proposed back in 1984 by Rivest and Shamir [RS84]. They constructed 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.[3] More recently, motivated by the task of securely pairing wireless devices (e.g., wireless USB or Bluetooth devices), this model was formalized by Vaudenay [Vau05] in the computational setting and extended by Naor et al. [NSS06, NSS08] to the statistical setting (considering computationally-bounded and computationally-unbounded adversaries, respectively).

Given that the out-of-band channel is of low bandwidth, it is of extreme importance to construct out-of-band authentication protocols with an essentially optimal tradeoff between the length of their out-of-band authenticated value and the adversary's success probability. Vaudenay and Naor et al. provided a complete characterization of this tradeoff, resulting in optimal computationally-secure and statistically-secure protocols.

[1] Despite the significant threats posed by man-in-the-middle attacks, research on the security of group messaging has so far assumed an initial authenticated setup phase (e.g., [CGCG$^+$17, RMS18]), and did not address this security-critical assumption.

[2] For example, as specified in WhatsApp's security whitepaper [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".

[3] Unfortunately, potential attacks on the Interlock protocol were identified later on [BM94, Ell96].

Out-of-band authentication: The group setting. Motivated by the insufficiently explored security of group messaging, we initiate the study of out-of-band message authentication protocols in the group setting. We extend the user-to-user setting to consider a group of users that consists of a sender (e.g., the group administrator) and multiple receivers (e.g., all other group members): All users communicate over an insecure channel, and we assume that the sender can out-of-band authenticate one short message to all receivers.[4] As in the user-to-user setting, this can be based, for example, on the assumption that each user can identify the administrator's voice, and having the administrator record and broadcast a short voice message. As above, we assume that an adversary may read or remove any message sent over the out-of-band channel for some or all receivers, and may delay it for different periods of time for different receivers, but cannot modify it in an undetectable manner.

Equipped with such an authentication protocol, the users of a group can now authenticate their public keys, or more generally, authenticate the transcript of any group key-exchange protocol of their choice. As in the user-to-user setting, given that the out-of-band channel is of low bandwidth, we aim at identifying the optimal tradeoff between the length of the out-of-band authenticated value and the adversary's success probability, and at constructing protocols that achieve this best-possible tradeoff.

1.1 Our Contributions

Modeling out-of-band authentication in the group setting. In this work we first put forward a realistic framework and strong notions of security for out-of-band message authentication protocols in the group setting. We consider a group of users that consists of a sender (e.g., the group administrator) and k receivers (e.g., all other group members), where for every $i \in [k]$ the sender would like to authenticate a message m_i to the ith receiver. We assume that all users are connected via an insecure channel (over which a man-in-the-middle adversary has complete control), and via a low-bandwidth "out-of-band" authenticated channel that enables the sender to authenticate one short message to all receivers. Adversaries may read or remove this message for some or all receivers, and may delay it for different periods of time for different receivers, but cannot modify it in an undetectable manner (we refer the reader to Section 3 for a formal description of our communication model and notions of security).

Identifying the optimal tradeoff: Protocols and matching lower bounds. Within our framework we then construct out-of-band authentication protocols with an optimal tradeoff between the length of their out-of-band authenticated value and the adversary's success probability. We consider both the computational setting where security is guaranteed against computationally-bounded adversaries, and the statistical setting where security is guaranteed against computationally-unbounded adversaries. In each setting we construct an authentication protocol, and then prove a lower bound showing that our protocol achieves essentially the best possible tradeoff between the length of the out-of-band authenticated value and the adversary's success probability. Our results are briefly summarized in Table 1, and we refer the reader the following section for a more detailed overview and theorem statements.

Computational vs. statistical security. Our tight bounds reveal a significant gap between the possible length of the out-of-band authenticated value in the computational setting and in the

[4] Clearly, one may consider a less-minimal extension where *several* users are allowed to send out-of-band authenticated values (i.e., not only the group administrator that we denote as the sender), but as our results show this is in fact not required.

	Our Protocols	**Our Lower Bounds**
Computational Security	$\boldsymbol{\log(1/\epsilon) + \log k}$	$\boldsymbol{\log(1/\epsilon) + \log k - \Theta(1)}$
Statistical Security	$\boldsymbol{(k+1) \cdot (\log(1/\epsilon) + \log k + \Theta(1))}$	$\boldsymbol{(k+1) \cdot \log(1/\epsilon) - k}$

Table 1: The length of the out-of-band authenticated value in our protocols and lower bounds. We denote by k the number of receivers (i.e., we consider groups of size $k+1$), and by ϵ the adversary's forgery probability. Our computationally-secure protocol relies on the existence of any one-way function (see Theorem 1.1), whereas our statistically-secure protocol and our two lower bounds do not rely on any computational assumptions (see Theorems 1.2, 1.3 and 1.4).
Note that our upper bound and lower bound in the computational setting match within an additive constant term, whereas in the statistical setting they match within an additive $(k+1)\log k + \Theta(k)$ term (however, whenever $\epsilon = o(1/k)$ as one would typically expect when setting a bound on the adversary's forgery probability, this difference becomes a lower-order term).

statistical setting: Whereas in the statistical setting we prove a lower bound that depends linearly on the size of the group, the length of the out-of-band authenticated value in our computationally-secure protocol depends very weakly on the size of the group. Moreover, when instantiating its cryptographic building block (a concurrent non-malleable commitment scheme) in the random-oracle model, our approach yields an efficient and practically-relevant protocol (which, alternatively, can also be based on any one-way function in the standard model).[5]

1.2 Overview of Our Contributions

A naive approach to constructing an out-of-band authentication protocol in the group setting is to rely on any such protocol in the user-to-user setting: Given a sender and k receivers, we can invoke a user-to-user protocol between the sender and each of the receivers. Thus, if the length of the out-of-band authenticated value in the underlying user-to-user protocol is $\ell(\epsilon)$ bits (where ϵ is the adversary's forgery probability), then the length of the out-of-band authenticated value in the resulting group protocol is $k \cdot \ell(\epsilon/k)$ bits.[6] Thus, the naive approach yields out-of-band authenticated values whose length is linear in the size of the group, and the key technical challenge underlying our work is understanding whether or not this is the best possible.

Concretely, the user-to-user protocols of Vaudenay [Vau05] and Naor et al. [NSS06] have out-of-band authenticated values of lengths $\ell(\epsilon) = \log(1/\epsilon)$ and $\ell(\epsilon) = 2\log(1/\epsilon) + \Theta(1)$, respectively. Thus, instantiating the naive approach with their protocols yields computationally-secure and statistically-secure protocols where the sender out-of-band authenticates $k \cdot (\log(1/\epsilon) + \log k)$ bits and $2k \cdot (\log(1/\epsilon) + \log k + \Theta(1))$ bits, respectively.

Our results show that, unlike in the user-to-user setting, in the group setting computationally-secure and statistically-secure protocols exhibit completely different behaviors. First, we show that for computationally-secure protocols it is possible to do dramatically better compared to the naive approach and completely eliminate the linear dependency on the size of the group. We prove the following two theorems providing an out-of-band authentication protocol and a matching lower bound:

[5] Concretely, when setting the adversary's forgery probability ϵ to 2^{-30} in a group that consists of $k = 2^{10}$ users, then in any statistically-secure protocol more than $k \cdot \log(1/\epsilon) = 2^{10} \cdot 30$ bits must be out-of-band authenticated, whereas in our computationally-secure protocol only $\log(1/\epsilon) + \log k = 40$ bits are out-of-band authenticated.

[6] Note that if the adversary's forgery probability in the group protocol should be at most ϵ, then the user-to-user protocol should be parameterized, for example, with ϵ/k as the adversary's forgery probability (enabling a union bound over the k executions).

Theorem 1.1. *Assuming the existence of any one-way function, for any $k \geq 1$ there exists a computationally-secure constant-round k-receiver out-of-band message authentication protocol in which the sender out-of-band authenticates $\log(1/\epsilon) + \log k$ bits, where ϵ is the adversary's forgery probability.*

Theorem 1.2. *In any computationally-secure k-receiver out-of-band message authentication protocol, the sender must out-of-band authenticate at least $\log(1/\epsilon) + \log k - \Theta(1)$ bits, where ϵ is the adversary's forgery probability.*

Then, we show that for statistically-secure protocols the naive approach is in fact asymptotically optimal, but it can still be substantially improved by a multiplicative constant factor (which is of key importance given that the out-of-band channel is of low bandwidth). We prove the following two theorems, once again providing an out-of-band authentication protocol and a lower bound:

Theorem 1.3. *For any $k \geq 1$ there exists a statistically-secure k-receiver out-of-band message authentication protocol in which the sender out-of-band authenticates $(k+1) \cdot (\log(1/\epsilon) + \log k + \Theta(1))$ bits, where ϵ is the adversary's forgery probability.*

Theorem 1.4. *In any statistically-secure k-receiver out-of-band message authentication protocol, the sender must out-of-band authenticate at least $(k + 1) \cdot \log(1/\epsilon) - k$ bits, where ϵ is the adversary's forgery probability.*

As discussed above, note that here our upper bound and lower bound differ by an additive $(k+1)\log k + \Theta(k)$ term. However, whenever $\epsilon = o(1/k)$ as one would typically expect when setting a bound on the adversary's forgery probability, this difference becomes a lower-order term.

In the remainder of this section we overview the main ideas underlying our protocols and lower bounds, first describing our contributions in the computational setting, and then describing our contributions in the statistical setting.

Computational security: Our protocol. Our computationally-secure protocol is inspired by the user-to-user protocol proposed by Vaudenay [Vau05]. In his protocol the sender S first commits to the value (m, r_S), where m is the message to be authenticated, and r_S is a random ℓ-bit string. The receiver R then replies with a random string r_R, followed by S revealing r_S and out-of-band authenticating $r_S \oplus r_R$. Finally, the receiver R accepts m if and only if the out-of-band authenticated value is consistent with his view of the protocol.

When moving to the group setting, however, a man-in-the-middle adversary has many more possible ways to interleave its interactions with the parties, thus providing security becomes a much more intricate task. For instance, a naive attempt to generalize Vaudenay's protocol to the group setting (while keeping the out-of-band authenticated value short) might naturally rely on the following idea: Have the sender choose a single value r_S and send each receiver a commitment to (m_i, r_S),[7] and then have each receiver R_i reply with a string r_{R_i} to all other parties.[8] The out-of-band authenticated value is then $r_S \oplus r_{R_1} \oplus \ldots \oplus r_{R_k}$, and each receiver R_i accepts the message m_i if and only if this value is consistent with his view of the protocol. Alas, this protocol is completely insecure – even when considering just one additional receiver. For example, an adversary can send R_1 a commitment to $(\widehat{m_1}, \widehat{r_S})$ for a message $\widehat{m_1} \neq m_1$ and an arbitrary $\widehat{r_S}$. After learning r_S and r_{R_2}, the adversary

[7] Of course, a commitment scheme may be interactive, but we use this terminology for ease of presentation in the overview.

[8] We do not go into details regarding the possible models of insecure communication in this high-level overview, and we refer the reader to Section 3 for an in-depth discussion.

can simply send R_1 the value $\widehat{r_{R_2}} = r_{R_2} \oplus r_S \oplus \widehat{r_S}$ instead of r_{R_2}. Since $r_S \oplus r_{R_2} = \widehat{r_S} \oplus \widehat{r_{R_2}}$, the attack will go undetected and the receiver R_1 will accept a fraudulent message $\widehat{m_1}$.

To immune our protocol from attacks as the one described above, the receivers in our protocol must avoid sending their random strings in the clear. Rather, they too send *commitments* of these strings at the beginning of the protocol. Informally, our protocol proceeds as follows: (1) Each R_i sends a commitment to a random ℓ-bit string r_{R_i}; (2) S chooses a random string r_S and sends a commitment to (m_i, r_S) to each R_i; (3) The receivers open their commitments; (4) S opens her commitments; (5) S out-of-band authenticates $r_S \oplus r_{R_1} \oplus \ldots \oplus r_{R_k}$. One can verify that the additional commitments indeed prevent the aforementioned attack, but there are clearly many additional attacks to consider given that an adversary has many possible ways to interleave its interactions with the parties.

The multitude of commitments in our protocol, and the many possible synchronizations an adversary may impose on them in the group setting, make proving the security of our protocol a challenging task. Nonetheless, we are able to show that when the commitment scheme being used is a concurrent non-malleable commitment scheme (see Section 2 for a formal definition), our protocol is indeed secure: Setting $\ell = \log(1/\epsilon) + \log k$ guarantees that the adversary's forgery probability is at most ϵ.

Technical details omitted, the intuition behind the security of the protocol is the following. An adversary A wishing to cause some R_i to accept a fraudulent message, essentially has to choose between two options. If A delivers all commitments to S and to R_i before R_i reveals r_{R_i}, then R_i accepting a fraudulent message implies breaking the concurrent non-malleability of the commitment scheme: The $2k$ commitments delivered to S and to R_i by the adversary must define values whose exclusive-or is equal to $r_{R_i} \oplus r_S$. These commitments thus satisfy a "non-trivial" relation which violates the concurrent non-malleability of the commitment scheme. On the other hand, if r_{R_i} is revealed before all commitments were delivered to S, then r_S is chosen after all commitments were delivered to S and to R_i. Hence, all other values contributing to the authenticated value sent by S, and to the value R_i is expecting to see as the out-of-band authenticated value, have already been determined, so the exclusive-or of all relevant values guarantees that the probability of the chosen r_S to result in equality is $2^{-\ell}$.

Computational security: Lower bound. Already in the user-to-user setting, at least $\log(1/\epsilon)$ bits must be out-of-band authenticated, where ϵ is the adversary's forgery probability. This can be proved, for example, by analyzing the collision entropy of the random variable corresponding to the out-of-band authenticated value (see, for example, [PV06]). We show that such an analysis can be extended to the group setting, resulting in a stronger lower bound which depends on the size of the group (and is in fact optimal given our above-described protocol).

Specifically, we show an efficient attack against any k-receiver protocol that succeeds with probability roughly $k \cdot 2^{-\ell}$, where ℓ is the number of bits the sender authenticates out-of-band. Given such a protocol π involving a sender and k receivers, our attacker runs $k+1$ independent executions of π, one with each party taking part in the protocol. In each execution, the attacker independently chooses k random messages as the input to the sender (the true sender in the execution with the sender, and the simulated one in the executions with each of the receivers), and honestly simulates the roles of all other parties. Now, if the out-of-band authenticated value in the execution with the sender is equal to the out-of-band authenticated value in one of the k executions with the receivers, then the attacker combines these two executions by forwarding the out-of-band authenticated value that is sent by the true sender for replacing the simulated value in the execution with that receiver.

Observe that the probability of a successful forgery is roughly the probability that the out-of-band authenticated value in the execution with the sender is indeed equal to the out-of-band authenticated value in one of the k executions with the receivers.[9] Hence, in order to analyze the effectiveness of this attack, it is sufficient to bound the probability of this event. We manage to provide a $\Theta(k \cdot 2^{-\ell})$ lower bound on the probability of this event, which yields Theorem 1.2.

Statistical security: Our protocol. The starting point of our statistically-secure protocol is the iterative hashing protocol of Naor et al. [NSS06]. Loosely speaking, in their protocol the parties maintain a joint sequence of values of decreasing length, starting with the input message of the sender and ending up with the out-of-band authenticated value. In each round, the parties apply to the current value a hash function that is *cooperatively chosen* by both parties: Half of the randomness for choosing the function is determined by the sender, and the other half by the receiver.

As noted above, when moving to the group setting, a naive generalization of the Naor et al. protocol in which the sender executes the user-to-user protocol with each receiver independently, will result in a blow-up of factor k in the length of the out-of-band authenticated value. However, we show that it is possible to exploit the specific structure of the Naor et al. protocol, and in particular of the out-of-band authenticated value, in order to cut its length in the group setting roughly by half (compared to the naive generalization). The main observation underlying our approach is that the k executions of the user-to-user protocol need not be completely independent. More concretely, we show that if in the last round (before sending the out-of-band authenticated value), the sender contributes *the same randomness* for all k hash functions, then all k executions are "tied together" in a way that permits a significant reduction in the number of bits that are authenticated out-of-band. Security is now of course not trivially guaranteed, as this change introduces heavy dependencies between the executions. We nevertheless manage to prove, carefully adjusting the structure of our protocol, that the resulting protocol provides an essentially optimal tradeoff between the length of the out-of-band authenticated value and its security.

Statistical security: Lower bound. We prove our lower bound in the statistical security setting by providing a lower bound on the Shanon entropy of the random variable corresponding to the out-of-band authenticated value in any out-of-band authentication protocol. Intuitively speaking, at the beginning of any such protocol, the out-of-band authenticated value is completely undetermined, while at the end of the execution it is fully determined. We show that if the forgery probability is to be bounded by ϵ, this decline in entropy must adhere to a specific structure: Each party must decrease the entropy of the out-of-band authenticated value – via the messages it sends during the execution of the protocol – by at least $\log(1/\epsilon) - 1$ bits on average. It follows that $\mathrm{H}(\Sigma) \geq (k+1) \cdot \log(1/\epsilon) - k$, where Σ is the afore-defined random variable and k is the number of receivers.

We formalize and prove this intuition by presenting a collection of $k+1$ attacks against any k-receiver out-of-band authentication protocol, one per each participating party. Loosely speaking, the attack corresponding to party P (where P may be the sender or any of the receivers) consists of running two executions of the protocol. First, our adversary plays the role of P in an honest execution of the protocol with all other parties, and obtains the out-of-band authenticated value σ to be sent at the end of this execution. Then, the adversary runs an execution of the protocol with P, playing the role of all other parties, while choosing their messages throughout the protocol not only conditioned on their views, but also conditioned on the out-of-band authenticated value being σ. We show in our analysis that if we denote by ϵ_P the success probability of the attack corresponding to party P, then it holds that $\prod_P \epsilon_P \geq 2^{-\mathrm{H}(\Sigma)-k}$. Hence, if the probability of a successful forgery in

[9] A successful forgery also requires that the input message for that particular receiver is different in the two executions, but this has little effect on the probability of forgery when the input messages are not too short.

any attack (and in particular in our $k+1$ attacks) is at most ϵ, then it holds that

$$2^{-\mathrm{H}(\Sigma)-k} \leq \prod_{P} \epsilon_P \leq \epsilon^{k+1},$$

and our lower bound follows. Our proof technique is inspired by the lower bound of Naor et al. [NSS06] for statistically-secure user-to-user out-of-band authentication protocols. In the group setting, however, there are many more "independent" attacks to consider, adding to the intricacy of the proof.

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we review the basic notions and tools that are used in this paper. In Section 3 we put forward our framework for out-of-band message authentication protocols in the group setting, formally discussing our communication models and notions of security. Then, in Sections 4 and 5 we present our protocols and prove our corresponding lower bounds in the computational and statistical settings, respectively.

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 $\{1,\ldots,n\}$. A function $\nu : \mathbb{N} \to \mathbb{R}^+$ 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 and mutual information. For random variables X, Y and Z we rely the following standard notions:

- The *entropy* of X is defined as $\mathrm{H}(X) = -\sum_x \Pr[X = x] \cdot \log \Pr[X = x]$.
- The *conditional entropy* of X given Y is defined as $\mathrm{H}(X|Y) = \sum_y \Pr[Y = y] \cdot \mathrm{H}(X|Y = y)$.
- The *mutual information* of X and Y is defined as $\mathrm{I}(X;Y) = \mathrm{H}(X) - \mathrm{H}(X|Y)$.
- The *mutual information* of X and Y given Z is defined as $\mathrm{I}(X;Y|Z) = \mathrm{H}(X|Z) - \mathrm{H}(X|Z,Y)$.

Non-malleable commitment schemes. In this paper we rely on the notion of statistically-binding concurrent 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. For simplicity, the definition below only addresses the one-many setting (which is equivalent to the general many-many setting [PR05]), as this is enough for our needs. Lin and Pass [LP11] and Goyal [Goy11] have shown that constant-round concurrent non-malleable commitment schemes can be constructed from any one-way function (the round complexity was further improved by Ciampi et al. [COS$^+$17] to just 4 rounds). From a more practical perspective, such schemes can be constructed efficiently in the random-oracle model [BR93]. For further information regarding non-malleable and concurrent non-malleable commitment schemes see, for example, [DDN00, CIO98, FF00, CF01, PR05, PR08, LPV08] and the references therein.

Intuitively speaking, a (one-many) concurrent non-malleable commitment scheme has the following guarantee: Any efficient adversary cannot use a commitment to some value v in order to produce commitments to values $\widehat{v_1},\ldots,\widehat{v_k}$ that are "non-trivially" related to v. More formally, Let $\mathsf{Com} = (C, R)$ be a statistically-binding commitment scheme, and let $k = k(\cdot)$ be a function of the security parameter $\lambda \in \mathbb{N}$, bounded by some polynomial. Consider an efficient adversary A that gets an auxiliary input $z \in \{0,1\}^*$ (in addition to the security parameter) and participates in the following "man-in-the-middle" experiment. A takes part in a single "left" interaction and in k "right" interactions: In the left interaction, A interacts with the committer C, and receives a commitment to a value v. Denote the resulting commitment (transcript of the interaction) by c. In the right interactions, A interacts with the receiver R, resulting in k commitments $\widehat{c_1},\ldots,\widehat{c_k}$. We define k related values $\widehat{v_1},\ldots,\widehat{v_k}$ in the following manner. For every $i \in [k]$, if $\widehat{c_i} = c$, if $\widehat{c_i}$ is not a valid commitment, or if $\widehat{c_i}$ can be opened to more than one value, we let $\widehat{v_i} = \perp$ (note that by the statistical binding property of Com, the latter case only happens with negligible probability). Otherwise, $\widehat{v_i}$ is the unique value to which $\widehat{c_i}$ may be opened. Let $\mathsf{mim}^{A}_{\mathsf{Com}}(v, z)$ denote the random variable that includes the values $\widehat{v_1},\ldots,\widehat{v_k}$ and A's view at the end of the afore-described experiment.

Definition 2.1. Let A and D be a pair of algorithms. We define the advantage of (A, D) with respect to security parameter $\lambda \in \mathbb{N}$ as

$$\mathsf{Adv}^{A,D}_{\mathsf{Com}}(\lambda) \overset{\mathsf{def}}{=} \max_{v,v' \in \{0,1\}^\lambda} \left\{ \Pr\left[D(1^\lambda, \mathsf{mim}^{A}_{\mathsf{Com}}(v, z)) = 1\right] - \Pr\left[D(1^\lambda, \mathsf{mim}^{A}_{\mathsf{Com}}(v', z)) = 1\right] \right\}.$$

We say that a statistically-binding commitment scheme is *concurrent non-malleable* if for any pair of probabilistic polynomial-time algorithms (A, D) there exists a negligible function $\nu = \nu(\cdot)$ such that $\mathsf{Adv}^{A,D}_{\mathsf{Com}}(\lambda) \leq \nu(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

3 The Communication Model and Notions of Security

We consider the message authentication problem in a setting involving a group of $k+1$ users: A sender S and k receivers $R_1,\ldots,R_k$. For each $i \in [k]$ the sender would like to authenticate a message m_i to the ith receiver R_i. We assume that the users communicate over two channels: An insecure channel over which a man-in-the-middle adversary has complete control, and a low-bandwidth "out-of-band" authenticated channel, enabling the sender to authenticate one short message to all receivers. In what follows we formally specify the underlying communication model as well as the notions of security that we consider in this work (generalizing those of Vaudenay [Vau05] and Naor et al. [NSS06] to the group setting).

3.1 Communication Model

Our starting point is the framework of Vaudenay [Vau05] and Naor et al. [NSS06] which considers a sender who wishes to authenticate a single message to a single receiver using out-of-band authentication. They modeled this interaction by providing the sender and the receiver with 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 (an "out-of-band" channel).

We extend this model to the group setting in the following manner. Similarly to the framework of Vaudenay and Naor et al. we assume that the parties are connected via two types of communication channels: Insecure channels and an authenticated low-bandwidth channel. As for the authenticated channel, we assume that the sender S is equipped with an out-of-band channel, through which S may send a short message visible to all receivers in an authenticated manner (e.g., a voice message in group messaging). The adversary may read or remove this message for some or all receivers, and may delay it for different periods of time for different receivers, but cannot modify it in an undetectable manner. One may also consider a scenario where S, as well as the receivers, may send multiple messages over the out-of-band authenticated channel throughout the protocol. However, this is less desirable from a practical standpoint, and in any case, will not be necessary in our protocols. Furthermore, our lower bounds readily capture this more general case as well, providing a lower bound on the total number of bits sent over the authenticated channel throughout the protocol.

As mentioned above, we also assume that the parties are connected among themselves in a network of insecure channels. These channels are vulnerable to man-in-the-middle attacks, and the adversary is assumed to have complete control over them: The adversary can read, delay and stop messages sent by the parties, as well as insert new messages 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 messages in the ith round of the protocol only upon receiving all due messages of round $i-1$.

One may consider various possible networks to define the topology of the insecure channels. Two extremes of that spectrum are the following:

- **The star network model:** In this model each receiver R_i is connected to the sender S via a bidirectional insecure channel. In particular, the receivers cannot send messages directly to each other, and any communication among them must pass through the sender S.

- **The complete network model:** In this model every pair of parties (sender and receiver as well as two receivers) is connected through an insecure channel.

In that respect, our results – both in the computational setting and in the statistical setting – will be of the strongest form possible. Our protocols will be articulated, and their correctness and security proven, in the restrictive "star" network model, which in particular means that they can be implemented in models richer in channels, and namely in the complete network model (in that case, some communication efficiency optimizations are possible). Our lower bounds on the other hand, will assume complete communication networks, and will hence apply to weaker network models as well.

3.2 Notions of Security

In what follows we define the security and correctness requirements of out-of-band authentication protocols, essentially extending those of Vaudenay [Vau05] and Naor et al. [NSS06] to the group setting in an intuitive manner. In such protocols, the input to the sender S is a vector of message $m_1, \ldots, m_k$ which may be chosen by the adversary. At the end of the execution, each receiver R_i outputs either a message $\widehat{m_i}$ or the unique symbol $\perp$, implying rejection. Informally, correctness states that in an honest execution, with high probability all receivers output the correct message; i.e., $\widehat{m_i} = m_i$ for every $i \in [k]$. As for security, we demand that an adversary (which is efficient in the computational setting and unbounded in the statistical setting) cannot convince a receiver to output an incorrect message; i.e., the probability that $\widehat{m_i} \notin \{m_i, \perp\}$ is bounded by a pre-specified parameter.

For the sake of generality, Definitions 3.1 and 3.2 below are articulated without specific reference to an underlying communication model, and may be applied to any of the group communication models discussed above. We begin with a formal definition of out-of-band authentication in the statistical setting.

Definition 3.1. A statistically-secure out-of-band $(n, \ell, k, r, \epsilon)$-authentication protocol is a $(k+1)$-party r-round protocol in which the sender S is invoked on a k-tuple of n-bit messages, and sends at most ℓ bits over the authenticated out-of-band channel. The following requirements must hold:

- **Correctness:** In an honest execution of the protocol, for all input messages $m_1, \dots, m_k \in \{0,1\}^n$ to S and for every $i \in [k]$, receiver R_i outputs m_i with probability 1.
- **Unforgeability:** For any adversary and for every adversarially-chosen input messages $m_1, \dots,$ m_k on which S is invoked, the probability that there exists some $i \in [k]$ for which receiver R_i outputs some message $\widehat{m_i} \notin \{m_i, \bot\}$ 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 $\lambda \in \mathbb{N}$.

Definition 3.2. Let $n = n(\lambda), \ell = \ell(\lambda), k = k(\lambda), r = r(\lambda)$ and $\epsilon = \epsilon(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. A computationally-secure out-of-band $(n, \ell, k, r, \epsilon)$-authentication protocol is a $(k+1)$-party r-round protocol in which the sender S is invoked on a k-tuple of n-bit messages, and sends at most ℓ bits over the authenticated out-of-band channel. The following requirements must hold:

- **Correctness:** In an honest execution of the protocol, for all input messages $m_1, \dots, m_k \in \{0,1\}^n$ to S and for every $i \in [k]$, receiver R_i outputs m_i with probability 1.
- **Unforgeability:** For any probabilistic polynomial-time adversary there exists a negligible function $\nu = \nu(\cdot)$ such that the following holds: For every input messages $m_1, \dots, m_k$ chosen by the adversary and on which S is invoked, the probability that there exists some $i \in [k]$ for which receiver R_i outputs some message $\widehat{m_i} \notin \{m_i, \bot\}$ is at most $\epsilon + \nu(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

4 The Computational Setting

In this section we prove tight bounds for computationally-secure out-of-band authentication in the group setting. In Section 4.1 we present our computationally-secure protocol and discuss its possible instantiations (both in the standard model and in the random-oracle model). In Section 4.2 we prove the security of our protocol, and in Section 4.3 we prove a matching lower bound on the length of the out-of-band authenticated value in any computationally-secure protocol.

4.1 Our Protocol and its Instantiations

Let $\mathsf{Com} = (C_{\mathsf{Com}}, R_{\mathsf{Com}})$ be a concurrent non-malleable commitment scheme that is statistically binding (see Section 2 and Definition 2.1). Our protocol, denoted π_{Comp}, is parameterized by the security parameter $\lambda \in \mathbb{N}$, by the number $k = k(\lambda)$ of receivers, by the length $\ell = \ell(\lambda)$ of the out-of-band authenticated value, and by the length $n = n(\lambda)$ of the messages that the user would like to authenticate. The protocol is defined as follows:

1. For every $i \in [k]$ the receiver R_i chooses a random ℓ-bit string $r_i \leftarrow \{0,1\}^\ell$, and commits to it to the sender S using Com. For every $i \in [k]$ denote the resulting commitment according to the view of R_i by c_i, and denote the commitments received by S by $\widehat{c_i}$.[10]

[10] 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.

2. The sender S chooses a random string $r_s \leftarrow \{0,1\}^\ell$, and executes k (possibly parallel) executions of Com to commit to the message (m_i, r_s) to the receiver R_i for every $i \in [k]$. Denote the resulting commitments, as seen by the sender S by c^i_s, and denote the commitment received by R_i by $\widehat{c^i_s}$.

   For every $i \in [k]$ the sender S also explicitly appends the following information to the first message it sends R_i as part of the commitment: (1) The message m_i, and (2) the (possibly tampered with) commitments $(\widehat{c_j})_{j\in[k]\setminus\{i\}}$ received from the other receivers in Step 1 of the protocol. We let $\widehat{m_i}$ and $(\widehat{c_{j\to i}})_{j\in[k]\setminus\{i\}}$ denote the message and the forwarded commitments as received by R_i.

3. For every $i \in [k]$ the receiver R_i sends a decommitment d_i of her commitment from Step 1 of the protocol to reveal r_i to the sender S. Let $\widehat{d_i}$ denote the decommitment received by S from R_i. For every $i \in [k]$ the sender S then checks whether $\widehat{d_i}$ is a valid decommitment to $\widehat{c_i}$. If so, let $\widehat{r_i}$ denote the committed value. Otherwise, S sends $\perp$ over the authenticated channel, in which case all receivers output $\perp$.

4. For every $i \in [k]$, the sender S sends receiver R_i a decommitment d^i_s to the corresponding commitment from Step 2 of the protocol, and reveals r_s to R_i. Denote by $\widehat{d^i_s}$ the decommitment received by R_i. For every $i \in [k]$ the receiver R_i checks if $\widehat{d^i_s}$ is a valid decommitment to $\widehat{c^i_s}$. If it is, denote the committed value by $(\widehat{m'_i}, \widehat{r^i_s})$. If it is not a valid decommitment or if $\widehat{m'_i} \neq \widehat{m_i}$ (where $\widehat{m_i}$ was received in Step 2), then R_i outputs $\perp$ and terminates.

   For every $i \in [k]$ the sender S also sends R_i the (possibly tampered with) decommitments $(\widehat{d_j})_{j\in[k]\setminus\{i\}}$ she received in Step 3. We let $(\widehat{d_{j\to i}})_{j\in[k]\setminus\{i\}}$ denote the decommitments received by R_i. If for some $j \in [k] \setminus \{i\}$ it holds that $\widehat{d_{j\to i}}$ is not a valid decommitment to $\widehat{c_{j\to i}}$ received by R_i is Step 2, then R_i outputs $\perp$ and terminates. Otherwise, denote by $(\widehat{r_{j\to i}})_{j\in[k]\setminus\{i\}}$ the values obtained by opening the commitments.

5. S computes $\sigma = r_s \oplus \widehat{r_1} \oplus \ldots \oplus \widehat{r_k}$ and sends σ over the authenticated out-of-band channel. Every receiver R_i computes $\widehat{\sigma_i} = \widehat{r^i_s} \oplus \widehat{r_{1\to i}} \oplus \ldots \oplus \widehat{r_{i-1\to i}} \oplus r_i \oplus \widehat{r_{i+1\to i}} \oplus \ldots \widehat{r_{k\to i}}$, and then outputs $\widehat{m_i}$ (received in Step 2) if $\widehat{\sigma_i} = \sigma$ and outputs $\perp$ otherwise.

The following theorem (when combined with the existence of a constant-round concurrent non-malleable statistically-binding commitment scheme based on any one-way function – see Section 2) implies Theorem 1.1 as an immediate corollary:

Theorem 4.1. *Let $k = k(\cdot), \ell = \ell(\cdot), r = r(\cdot)$ and $n = n(\cdot)$ be functions of the security parameter $\lambda \in \mathbb{N}$ and let* Com *be an r-round concurrent non-malleable commitment scheme. Then, protocol π_{Comp} is a computationally-secure out-of-band $(n, \ell, k, O(r), k \cdot 2^{-\ell})$-authentication protocol.*

Possible instantiations. Our protocol π_{Comp} can be instantiated with Com being any concurrent non-malleable statistically-biding commitment scheme. From a theoretical point of view, Lin and Pass [LP11] and Goyal [Goy11] gave constant-round constructions of such schemes from any one-way function (and the round complexity was further improved by [COS$^+$17]). Hence, our protocol can also be instantiated as a constant-round protocol, assuming only the existence of one-way functions. This assumption is minimal and necessary, since Naor et al. [NSS06] showed that even in the user-to-user setting, any computationally-secure out-of-band authentication protocol for which $\ell < 2\log 1/\epsilon - \Theta(1)$ implies the existence of one-way functions.

From a more practical standpoint, a *non-interactive* concurrent non-malleable statistically-biding commitment scheme can be very efficiently constructed in the random oracle model [BR93]. Thus, instantiating π_{Comp} with a cryptographic hash function (e.g., SHA-2) as the random oracle yields a highly efficient protocol. Given a random oracle H, in order to commit to a value v, one simply has to send $c = H(v, r)$ for a sufficiently long random string r. Decommitment is done by revealing v and r, and the receiver asserts that $c = H(v, r)$. Consider a pair of poly-query algorithms (A, D), where A is the man-in-the-middle adversary and D is the distinguisher (see Definition 2.1). Informally speaking, assume H is sufficiently length-increasing (say, length-doubling) so that it is difficult to find an element y in its image without querying H on a pre-image of y. So the algorithm A, that receives $c = H(v, r)$ and produces $c_1 = H(v_1, r_1), \ldots, c_k = H(v_k, r_k)$, knows $v_1, \ldots, v_k$ with overwhelming probability. Hence, it can distinguish between the case that $c = H(v, r)$, and the case that $c = H(v', r')$ where the value v' – when taken together with $v_1, \ldots, v_k$ and the view of A – does not satisfy the polynomial time relation defined by the distinguisher D. By a standard argument, this is hard for any adversary making a polynomial number of queries to the random oracle.

Non-malleable commitment schemes also exist in the common reference string (CRS) model (see, for example, [CIO98, CKO$^+$01, FF00, CF01, DG03]). However, assuming a trusted CRS may be somewhat incompatible with the ad-hoc nature of instant messaging platforms and applications.

4.2 Proof of Security

In this section, we prove the unforgeability of protocol π_{Comp}, proving Theorem 4.1. For simplicity of presentation, we start by assuming that the commitment scheme Com, with which the protocol is instantiated, is a non-interactive commitment scheme (in addition to being statistically-binding and concurrent non-malleable), as this already captures the gist of the proof. We then discuss how to make the proof compatible with multiple-round commitment schemes.

Let $\lambda \in \mathbb{N}$ be the security parameter, and let $k = k(\lambda), \ell = \ell(\lambda)$ and $n = n(\lambda)$. For an adversary A, let $\mathsf{Forge}_{A,i}$ denote the event in which A fools R_i (i.e., R_i outputs $\widehat{m_i} \notin \{m_i, \bot\}$) in an execution of π_{Comp} and let $\mathsf{Forge}_A = \bigcup_{i \in [k]} \mathsf{Forge}_{A,i}$. The following Lemma captures the unforgeability of π_{Comp}.

Lemma 4.2. *For any probabilistic polynomial-time adversary A, there exists a negligible function $\nu = \nu(\cdot)$ such that*

$$\Pr\left[\mathsf{Forge}_A\right] \leq k \cdot 2^{-\ell} + \nu(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$.

Proof. We will prove that for every efficient adversary A and for every $i \in [k]$ there exists a negligible function $\nu'(\cdot)$ such that $\Pr\left[\mathsf{Forge}_{A,i}\right] \leq 2^{-\ell} + \nu'(n)$. The Lemma will then follow by taking a union bound over all $i \in [k]$.

For a message m sent in the execution of the protocol, denote by $T(m)$ the time in which m was sent. By the definition of the protocol, for every $i \in [k]$, it always holds that $T(c_i) < T\left(\widehat{c_s^i}\right) < T(d_i) < T\left(\widehat{d_s^i}\right)$, as well as $T\left(\widehat{c_i}\right) < T\left(c_s^i\right) < T\left(\widehat{d_i}\right) < T\left(d_s^i\right)$. We also assume the following assumptions without loss of generality:

1. Whenever a party (either S or one of the receivers) is due to send a message according to the protocol, the adversary waits until this message is sent before deciding on its next action. In particular, for instance, this implies $T(c_i) < T\left(\widehat{c_j}\right)$ for every $i, j \in [k]$.

2. The adversary delays sending $\widehat{d_s^i}$ until the end of the execution; i.e., $T\left(d_s^i\right) < T\left(\widehat{d_s^i}\right)$.

Note that these two assumptions do not hamper a potential adversary's efforts, as any adversary can be turned into an adversary that is consistent with the above assumptions without damaging the probability of a successful forgery.

We now turn to bound the probability of such a forgery. First, consider the event in which at least one of the commitments in the execution (either a commitment that was sent by the adversary A or by one of the honest parties) does not uniquely define (information-theoretically speaking) a committed value v, and denote this event by Coll. By the statistical binding property of Com, there exists a negligible function ν_1 such that $\Pr\left[\mathsf{Coll}\right] \leq \nu_1(n)$. So for the remainder of the proof we will condition our analysis on the event $\overline{\mathsf{Coll}}$. We differentiate in our analysis between two cases: The event that $T\left(c_s^i\right) < T\left(d_i\right)$, which we denote by Sync_i, and that of $T\left(c_s^i\right) > T\left(d_i\right)$, which will be denoted by $\overline{\mathsf{Sync}_i}$. The bound thus follows by the following two claims. Claim 4.3 bounds the probability of forgery in case event $\overline{\mathsf{Sync}_i}$ occurs, and Claim 4.4 bounds it in case of event Sync_i.

Intuitively speaking, in case event $\overline{\mathsf{Sync}_i}$ occurs, S chooses r_s after the other variables that contribute to the authenticated string σ and to the string $\widehat{\sigma_i}$ (which R_i is expecting to see over the authenticated channel) have all already been fixed. Hence, there is exactly one value for r_s that will result in R_i accepting, which is obtained with probability $2^{-\ell}$. This intuition is captured by the following claim.

Claim 4.3. *For any adversary A and for any $i \in [k]$, it holds that*

$$\Pr\left[\mathsf{Forge}_{A,i} | \overline{\mathsf{Sync}_i} \wedge \overline{\mathsf{Coll}}\right] \leq 2^{-\ell}.$$

Proof of Claim 4.3. Fix $i \in [k]$. By the definition of the protocol, it holds that $T\left(c_s^i\right) > T\left(c_i\right)$ and $T\left(c_s^i\right) > T\left(\widehat{c_j}\right)$ for every $j \in [k] \setminus \{i\}$; i.e., c_s^i is sent after R_i sends c_i and after S receives $\widehat{c_j}$. Conditioned on $\overline{\mathsf{Sync}_i}$, it also holds that $T\left(c_s^i\right) > T(\widehat{c_s^i})$ and $T\left(c_s^i\right) > T\left(\widehat{c_{j\to i}}\right)$ for every $j \in [k] \setminus \{i\}$. Moreover, conditioned on $\overline{\mathsf{Coll}}$, each of these commitments can uniquely be opened to (at most) a single value. Meaning, in that case, the values $r_i, \widehat{r_s}$ and $\widehat{r_j}$ and $\widehat{r_{j\to i}}$ for every $j \in [k] \setminus \{i\}$ are determined before c_s^i is sent. Since S chooses r_s just before the sending c_s^i, and once the above listed values are already fixed (recall Step 2 of the protocol), it holds that

$$\begin{aligned}
\Pr\left[\mathsf{Forge}_{A,i} | \overline{\mathsf{Sync}} \wedge \overline{\mathsf{Coll}}\right] &\leq \Pr\left[r_s \oplus \left(\bigoplus_{j\in[k]} \widehat{r_j}\right) = \widehat{r_s} \oplus r_i \oplus \left(\bigoplus_{j\in[k]\setminus\{i\}} \widehat{r_{j\to i}}\right)\right] \\
&= \Pr\left[r_s = \left(\bigoplus_{j\in[k]} \widehat{r_j}\right) \oplus \widehat{r_s} \oplus r_i \oplus \left(\bigoplus_{j\in[k]\setminus\{i\}} \widehat{r_{j\to i}}\right)\right] \\
&= 2^{-\ell}.
\end{aligned}$$

■

The following claim shows that for any $i \in [k]$, conditioned on Sync_i, any adversary that breaks the unforgeability of π_{Comp} with respect to R_i can be transformed into an adversary breaking the concurrent non-malleability of Com.

Claim 4.4. *For any probabilistic polynomial-time adversary A and for any $i \in [k]$, there exists a pair of probabilistic polynomial-time algorithms (A_{NM}, D) such that*

$$\mathsf{Adv}_{\mathsf{Com}}^{A_{\mathsf{NM}},D} \geq \Pr\left[\mathsf{Forge}_{A,i} | \mathsf{Sync}_i \wedge \overline{\mathsf{Coll}}\right] - 2^{-\ell}.$$

We start by giving an intuitive overview of the reduction, and then continue to the formal proof. In broad brush strokes, A_{NM} gets as an auxiliary input a random value $z \leftarrow \{0,1\}^\ell$, and runs a single left interaction with a committer C_{Com} and $2k$ right interactions with a receiver R_{Com}. In the left interaction, C_{Com} sends a commitment c to a value $v \in \{0,1\}^\ell$. A_{NM} runs A and passes forward c to A as the commitment of R_i to the value r_i in Step 1 of the protocol. A_{NM} also simulates honestly the sender and all other receivers by itself (choosing random ℓ-bit strings and producing commitments/decommitments when necessary). Indeed, when A expects to receive the decommitment to c, the simulation will get "stuck", as A_{NM} does not know how to open c. However, conditioned on Sync_i, by the time A expects to receive the decommitment to c, A and A_{NM} have already issued commitments to all other values that make up σ and $\widehat{\sigma}_i$, yielding enough information to break the concurrent non-malleability of Com. Consider two cases: When $v = z$, the view of A is exactly as in an execution of π_{Comp} up to $T(d_i)$, and a distinguisher D that gets as input A_{NM}'s view as well as $\{\widehat{r_j}\}_{j\in[k]}, \{\widehat{r_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ and $\widehat{r_s^i}$ can check if $\sigma = \widehat{\sigma}_i$ (where r_i is substituted with z when computing $\widehat{\sigma}_i$), and the probability for that will be at least that of $\mathsf{Forge}_{A,i}$. On the other hand, when v is fixed to be some arbitrary value (e.g., 0^ℓ), the commitments issued by A_{NM} and by A are independent of z, and hence the probability that $\sigma = \widehat{\sigma}_i$ is $2^{-\ell}$. We now move on to the formal proof.

Proof of Claim 4.4. Fix $i \in [k]$. On input 1^λ and auxiliary input z, the adversary A_{NM} is defined by the following steps:

1. In the left interaction, A_{NM} gets a commitment c to a value v.
2. A_{NM} chooses k random ℓ-bit strings $r_s \leftarrow \{0,1\}^\ell$ and $r_j \leftarrow \{0,1\}^\ell$ for every $j \in [k] \setminus \{i\}$. It then runs $A(1^\lambda)$, and simulates a partial execution of π_{Comp} in the following manner:
   (a) A_{NM} forwards c to A as the commitment of R_i to value r_i. For every $j \in [k] \setminus \{i\}$, A_{NM} computes a commitment c_j for r_i and a corresponding decommitment d_j according to Com, and sends c_j to A as the commitment of R_j in the Step 1 of the protocol.
   (b) After A outputs a commitment $\widehat{c_j}$ for every $j \in [k]$, along with a message m_j as the input message to S to be authenticated to R_j, A_{NM} replies (simulating S) as follows. For every $j \in [k]$, A_{NM} computes a commitment c_s^j to (m_j, r_s) and a corresponding decommitment d_s^j according to Com. A_{NM} then sends the message $\left(c_s^j, m_j, (\widehat{c}_t)_{t\in[k]\setminus\{j\}}\right)$ as the message from S to R_j in step 2 of the protocol, for every $j \in [k]$.
   (c) Similarly, in each of the interactions of π_{Comp}, when A outputs the tuple $\left(\widehat{c_s^j}, \widehat{m_j}, (\widehat{c_{t\to j}})_{t\in[k]\setminus\{j\}}\right)$ as the message delivered to R_j in Step 2 of the protocol, A_{NM} differentiates between two cases:
      - If $j \neq i$, then A_{NM} replies (simulating R_j) with d_j as computed in Step (a) of the simulation.
      - If $j = i$, A terminates the simulation.
   (d) In each interaction of π_{Comp} other then with receiver R_i, when A outputs $\widehat{d_j}$ as the decommitment received by S from R_j in Step 3 of π_{Comp}, A_{NM} replies (simulating S) with d_s^j computed in Step (b) of the simulation.
3. When the simulation is terminated, A_{NM} sends $\widehat{c_s^i}$, $\{\widehat{c_j}\}_{j\in[k]}$ and $\{\widehat{c_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ to R_{Com} as the commitments in its right interactions.

The distinguisher D gets as input the random variable $\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(v, z)$, which includes in particular the following variables:

- The auxiliary input z.
- The random string r_s drawn by A in the beginning of the simulation.
- The values $\{\widehat{r_j}\}_{j\in[k]}, \{\widehat{r_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ and $\widehat{r_s^i}$ which are the values to which the commitments $\{\widehat{c_j}\}_{j\in[k]}, \{\widehat{c_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ and $\widehat{c_s^i}$ may be opened to, respectively, or $\perp$. First, note that since we are conditioning on the event Sync_i, all these commitments were indeed produced by the simulated adversary A, before A_{NM} had to terminate the simulation. Furthermore, conditioned on $\overline{\mathsf{Coll}}$, each of the values D gets is $\perp$ either in the case that the corresponding commitment is not a valid one (and cannot be opened to any value) or that it is identical to c (the commitment of v received in the left interaction). In the latter case, again by conditioning of $\overline{\mathsf{Coll}}$, this means that the only value this commitment might be opened to is v.

On input the security parameter 1^λ and $\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(v, z)$, D is defined as follows:

1. If $\widehat{r_s^i} = \perp$, output 0.
2. Otherwise, let par_S denote the parity of the number of $\perp$ symbols among the values $\{\widehat{r_j}\}_{j\in[k]}$. That is, $\mathsf{par}_S = 1$ if the number of $\perp$ symbols among these values is odd, and $\mathsf{par}_S = 0$ otherwise. Let par_i be defined similarly as the parity of $\perp$ symbols among $\{\widehat{r_{j\to i}}\}_{j\in[k]\setminus\{i\}}$.

   D then computes $\sigma = r_s \oplus \left(\bigoplus_{j\in[k]} \widehat{r_j}\right) \oplus (\mathsf{par}_S \cdot z)$ and $\widehat{\sigma_i} = \widehat{r}_{si} \oplus z \oplus \left(\bigoplus_{j\in[k]\setminus\{i\}} \widehat{r_{j\to i}}\right) \oplus (\mathsf{par}_i \cdot z)$, where $\mathsf{par}_S \cdot z = 0^\ell$ if $\mathsf{par}_S = 0$ and $\mathsf{par}_S \cdot z = z$ if $\mathsf{par}_S = 1$.[11] Finally, D outputs 1 if $\sigma = \widehat{\sigma_i}$, and 0 otherwise.

For the analysis of the attack, consider two cases:

Case 1: When $v = z$ (in particular, v is also a uniformly chosen ℓ-bit string), A_{NM} perfectly simulates protocol π_{Comp} to A until it terminates. Conditioned on $\overline{\mathsf{Coll}}$, by the time the simulation terminates, A cannot open the commitments it sent in a way that will result in (the simulated) S sending a string other than σ (as computed by D) over the authenticated channel, or in (the simulated) R_i expecting a string other than $\widehat{\sigma_i}$ (as computed by D). It follows that only when the authentication values computed by D are equal (i.e., $\sigma = \widehat{\sigma_i}$), can the execution of π_{Comp} be completed such that R_i does not output $\perp$. In particular, this means that D outputting 1 is a necessary condition for $\mathsf{Forge}_{A,i}$. This yields

$$\Pr\left[D(1^\lambda, \mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(z, z)) = 1\right] \geq \Pr\left[\mathsf{Forge}_{A,i} \middle| \mathsf{Sync}_i \wedge \overline{\mathsf{Coll}}\right].$$

Case 2: When $v = 0^\ell$, all values that go into the computation of σ and $\widehat{\sigma_i}$ by D are independent of z. Namely, the view of the simulated adversary A is independent of z, as z does not affect the

[11] This has the effect of replacing every $\perp$ symbol in $\{\widehat{r_j}\}_{j\in[k]}$ and $\{\widehat{r_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ with z when computing σ and $\widehat{\sigma_i}$, respectively.

computation of any of the messages A_{NM} "feeds" it. Thus, it holds that

$$\begin{aligned}
&\Pr\left[D(\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(0^\ell, z)) = 1\right] \\
&\qquad = \Pr\left[z = r_s \oplus \left(\bigoplus_{j\in[k]} \widehat{r_j}\right) \oplus (\mathsf{par}_S \cdot z) \oplus \widehat{r_s^i} \oplus \left(\bigoplus_{j\in[k]\setminus\{i\}} \widehat{r_{j\to i}}\right) \oplus (\mathsf{par}_i \cdot z)\right] \\
&\qquad = 2^\ell.
\end{aligned}$$

Combining the above cases, it holds that

$$\Pr_{z\leftarrow\{0,1\}^\ell}\left[\Pr\left[D(\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(z, z)) = 1\right] - \Pr\left[D(\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(0^\ell, z)) = 1\right]\right] \geq \Pr\left[\mathsf{Forge}_i | \mathsf{Sync} \wedge \overline{\mathsf{Coll}}\right] - 2^\ell.$$

By an averaging argument, there exists some $z^* \in \{0,1\}^\ell$ such that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{A_{NM},D}}_{\mathsf{Com}} &\geq \Pr\left[D(\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(z^*, z^*)) = 1\right] - \Pr\left[D(\mathsf{mim}^{A_{\mathsf{NM}}}_{\mathsf{Com}}(0^\ell, z^*)) = 1\right] \\
&\geq \Pr\left[\mathsf{Forge}_i | \mathsf{Sync} \wedge \overline{\mathsf{Coll}}\right] - 2^\ell.
\end{aligned}$$

and the claim follows. ■

By Claim 4.4 and the concurrent non-malleability of Com, for every probabilistic polynomial-time adversary A, there exists a negligible function $\nu_2(\cdot)$ such that for every $i \in [k]$,

$$\Pr\left[\mathsf{Forge}_{A,i} | \mathsf{Sync}_i \wedge \overline{\mathsf{Coll}}\right] \leq 2^{-\ell} + \nu_2(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$. Combined with Claim 4.3 and the statistical binding property of Com, this yields that

$$\begin{aligned}
\Pr\left[\mathsf{Forge}_{A,i}\right] \leq\ & \Pr\left[\mathsf{Coll}\right] + \Pr\left[\mathsf{Forge}_{A,i} | \mathsf{Sync}_i \wedge \overline{\mathsf{Coll}}\right] \cdot \Pr\left[\mathsf{Sync}_i\right] \\
& + \Pr\left[\mathsf{Forge}_{A,i} | \overline{\mathsf{Sync}_i} \wedge \overline{\mathsf{Coll}}\right] \cdot \Pr\left[\overline{\mathsf{Sync}_i}\right] \\
\leq\ & \nu_1(n) + 2^{-\ell} + \nu_2(n).
\end{aligned}$$

Taking a union bound over all $i \in [k]$ and denoting $\nu(n) = k \cdot (\nu_1(n) + \nu_2(n))$, it holds that

$$\Pr\left[\mathsf{Forge}_A\right] \leq k \cdot 2^{-\ell} + \nu(n).$$

This concludes the proof of Lemma 4.2. ■

Supporting interactive commitments. The proof of Lemma 4.2 can be easily extended to support the use of concurrent non-malleable commitment schemes which are interactive. First, let's observe that any such commitment scheme can be converted into a scheme in which the sender's first message is in and of itself statistically biding; i.e., with overwhelming probability, once this message is sent, even a computationally unrestricted adversary cannot complete the commitment phase in two different ways, generating two different commitments c_1 and c_2, such that it is possible to open them into two distinct messages. This can be done by first having the sender committing to the message using Naor's statistically-binding two-message commitment scheme [Nao91], and then committing to the same message using the original non-malleable scheme. It is not hard to see that this transformation achieves the afore-described property, while retaining non-malleability that is sufficient for our needs; i.e., an efficient adversary cannot produce a commitment for a related yet different value.[12] Further observe that we can have each of the receivers $R_1, \dots, R_k$ append a random string of the appropriate length to its last message to S in Step 1 of the protocol, to be used as the first message of Naor's scheme, when S commits to the messages $\{(m_i, r_s)\}_{i\in[k]}$ in Step 2. Meaning, S waits until completing all commitments $c_1, \dots, c_k$ of stage 1, and only then chooses r_s and initiates the commitments of stage 2 in parallel, by sending a "binding message" to each receiver.

Now, for a commitment c, we denote by $T(c)$ the time of the first message sent by the committer as part of the commitment. In particular this means that for every $i \in [k]$, $T(c_s^i)$ is the time the first message is sent by S to R_i in Step 2 of the protocol. The event Sync_i, in which $T(c_s^i) < T(d_i)$, now captures executions in which d_i is sent after the first *and "binding"* message of c_s^i. Under this notation, Claim 4.3 still holds as conditioned on $\overline{\mathsf{Sync}_i}$, it is still true that r_s is sampled after all other variables contributing to the computation of σ and of $\widehat{\sigma_i}$ are uniquely determined. Claim 4.4 still holds since it is still the case that conditioned on Sync_i, the distinguisher D has sufficient information to compute σ and $\widehat{\sigma_i}$ by the time d_i is due in the simulation of π_{Comp} described in the proof.

A final, rather technical modification to the proof, is that in the proof of Claim 4.4, the adversary A_{NM} cannot simply complete the (partial) simulation of the protocol and then output $\widehat{c_s^i}$, $\{\widehat{c_j}\}_{j\in\{k\}}$ and $\{\widehat{c_{j\to i}}\}_{j\in[k]\setminus\{i\}}$ as the commitments in the right interactions. Rather, it has to execute the commitments in the right interactions *in parallel* to the aforementioned commitments, which the adversary A issues in the simulation (simply forwarding A's corresponding messages to R_{Com} as they come, and vice versa).

4.3 Lower Bound

In this section, we prove a lower bound on the length of out-of-band authenticated value in any out-of-band authentication protocol, as a function of the desired security level ϵ and of the number of receivers k. Our bound shows that the length of the out-of-band authenticated value in our protocol π_{Comp} of Section 4.1 is optimal (up to an additive constant). The lower bound is stated by the following Theorem, which yields Theorem 1.2.

Theorem 4.5. *For any computationally-secure $(n, \ell, k, r, \epsilon)$-authentication protocol where $n \geq \log(1/\epsilon) + \log k + 3$ and $\epsilon < 1/6$, it holds that $\ell \geq \log 1/\epsilon + \log k - 3$.*

Proof. Let $\pi = (S, R_1, \dots, R_k)$ be a k-receiver out-of-band authentication protocol for messages of length n in the complete network communication model. We present an efficient adversary A that succeeds in fooling at least one of the reveivers with probability at least $k \cdot 2^{-\ell-3}$, and the theorem follows (for an intuitive overview of the attack and analysis, see Section 1.2).

On input 1^λ, A runs the following steps:

1. A samples k input messages $(m_1, \dots, m_k) \leftarrow \{0,1\}^{m\times k}$ as the input to the sender S, and runs an execution with S in which A plays the role of all receivers. Denote by $\sigma \in \{0,1\}^\ell$ the value that S sends over the authenticated channel at the end of this execution.

2. For every $i \in [k]$, A samples k input messages $(\widehat{m_1^i}, \dots, \widehat{m_k^i}) \leftarrow \{0,1\}^{m\times k}$ uniformly at random (independently from the messages sampled in the other executions), and runs an execution of π with R_i in which A plays the role of the sender (with input $(\widehat{m_1^i}, \dots, \widehat{m_k^i})$) and all other

[12] The adversary might be able to produce a different commitment for the *same* value, but it is easy to see that this is benign in our protocol, and our proof of security still holds.

receivers. For every $i \in [k]$ denote the out-of-band authenticated value the (simulated) sender sends in the end of the execution with the true receiver R_i by $\widehat{\sigma_i}$.

We first wish to lower bound the probability that there exists some receiver R_i that outputs $\widehat{m_i^i}$. By the correctness of π, this is at least the probability that $\widehat{\sigma_i} = \sigma$. Thus, for every $i \in [k]$, it holds that

$$\Pr\left[R_i \text{ outputs } \widehat{m_i^i}\right] \geq \Pr\left[\widehat{\sigma_i} = \sigma\right] = \sum_{v \in \{0,1\}^\ell} \Pr\left[\sigma = v\right] \cdot \Pr\left[\widehat{\sigma_i} = v\right].$$

More generally, for any subset $\mathcal{I} \subseteq [k]$ of the receivers, it holds that

$$\begin{aligned}\Pr\left[\forall i \in \mathcal{I} : R_i \text{ outputs } \widehat{m_i^i}\right] &\geq \sum_{v \in \{0,1\}^\ell} \Pr\left[\sigma = v\right] \cdot \prod_{i \in \mathcal{I}} \Pr\left[\widehat{\sigma_i} = v\right] \\ &= \sum_{v \in \{0,1\}^\ell} \left(\Pr\left[\sigma = v\right]\right)^{|\mathcal{I}|+1}.\end{aligned}$$

The inequality follows by the fact that the executions A conducts with the receivers are independent from each other, and the equality holds since σ and $\widehat{\sigma_i}$ are identically distributed for every $i \in [k]$. The inclusion-exclusion principle now yields that the probability that for at least one receiver it holds that $\widehat{\sigma_i} = \sigma$ is

$$\Pr\left[\exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] \geq \sum_{i=1}^{k} (-1)^{i+1} \cdot \binom{k}{i} \cdot \left(\sum_{v \in \{0,1\}^\ell} \left(\Pr\left[\sigma = v\right]\right)^{i+1}\right).$$

The above probability is minimized when the distribution of σ over a random execution of the protocol as described above is uniform; i.e., when $\Pr\left[\sigma = v\right] = 2^{-\ell}$ for all $v \in \{0,1\}^\ell$. Hence, it holds that

$$\Pr\left[\exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] \geq \sum_{i=1}^{k} (-1)^{i+1} \cdot \binom{k}{i} \cdot 2^{-i \cdot \ell}.$$

In what follows, we make use of the following claim which bounds the above expression.

Claim 4.6. $\sum_{i=1}^{k} (-1)^{i+1} \cdot \binom{k}{i} \cdot 2^{-i \cdot \ell} \geq \min\left\{1/3, k \cdot 2^{-\ell}/4\right\}$.

Proof of Claim 4.6. In order to bound this expression, we differentiate between two cases. First, consider the case where $k \geq 3 \cdot 2^{\ell-1}$. The above expression can be thought of in the following manner: k balls are independently thrown into 2^ℓ bins uniformly at random. Let B be the random variable denoting the number of balls in the first bin at the end to the experiment. Then, the expression we wish to bound is exactly $\Pr\left[B > 0\right]$. Let N be a geometric random variable denoting the number of balls thrown until a ball hits the first bin.[13] Then, by Markov's bound it holds that

$$\begin{aligned}\Pr\left[\exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] &\geq \Pr\left[B > 0\right] \\ &= \Pr\left[N \leq k\right] \\ &= 1 - \Pr\left[N > k\right] \\ &\geq 1 - \frac{\mathbb{E}\left[N\right]}{k} \\ &\geq 1 - \frac{2^\ell}{3 \cdot 2^{\ell-1}} = \frac{1}{3}.\end{aligned}$$

[13] For the sake of defining N, balls are thrown until a ball is thrown into the first bin.

Now consider the case where $k < 3 \cdot 2^{\ell-1}$. In this case, it holds that

$$\begin{aligned}
&\Pr\left[\exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] \\
&\qquad \geq \sum_{i=1}^{k} (-1)^{i+1} \cdot \binom{k}{i} \cdot 2^{-i\cdot\ell} \\
&\qquad \geq \sum_{i=1}^{\lfloor k/2 \rfloor} \left(\binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell} - \binom{k}{2i} \cdot 2^{-2i\cdot\ell} \right) \\
&\qquad = \sum_{i=1}^{\lfloor k/2 \rfloor} \left(1 - \frac{\binom{k}{2i} \cdot 2^{-2i\cdot\ell}}{\binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell}} \right) \cdot \binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell} \\
&\qquad = \sum_{i=1}^{\lfloor k/2 \rfloor} \left(1 - \frac{k-2i+1}{2i} \cdot 2^{-\ell} \right) \cdot \binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell} \\
&\qquad > \sum_{i=1}^{\lfloor k/2 \rfloor} \left(1 - \frac{k}{2} \cdot 2^{-\ell} \right) \cdot \binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell} \\
&\qquad > \frac{1}{4} \cdot \sum_{i=1}^{\lfloor k/2 \rfloor} \binom{k}{2i-1} \cdot 2^{-(2i-1)\cdot\ell} > \frac{k}{4} \cdot 2^{-\ell}.
\end{aligned}$$

■

Let Forge_A denote the event in which R_i outputs $\widehat{m_i^i} \neq m_i$. By Claim 4.6,

$$\begin{aligned}
\Pr\left[\mathsf{Forge}_A\right] &= \Pr\left[\exists i \in [k] \text{ s.t. } m_i \neq \widehat{m_i^i} \wedge R_i \text{ outputs } \widehat{m_i^i}\right] \\
&\geq \Pr\left[\forall j \in [k], m_j \neq \widehat{m_j^j} \wedge \exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] \\
&\geq \Pr\left[\exists i \in [k] \text{ s.t. } R_i \text{ outputs } \widehat{m_i^i}\right] - \Pr\left[\exists j \in [k] \text{ s.t. } m_j = \widehat{m_j^j}\right] \\
&\geq \min\left\{\frac{1}{3}, \frac{k}{4} \cdot 2^{-\ell}\right\} - k \cdot 2^{-n} \\
&\geq \min\left\{\frac{1}{6}, k \cdot 2^{-\ell-2} - k \cdot 2^{-n}\right\}.
\end{aligned}$$

The last inequality holds since $n \geq \log k + \log 1/\epsilon + 3 > \log k + 3$ and thus $k \cdot 2^{-n} < 1/6$. Finally, since $\epsilon < 1/6$ and $n \geq \log k + \log 1/\epsilon$, it holds that

$$\epsilon \geq k \cdot 2^{-\ell-2} - k \cdot 2^{-n} \geq k \cdot 2^{-\ell-2} - \epsilon.$$

Equivalently, $\epsilon \geq k \cdot 2^{-\ell-3}$, which implies $\ell \geq \log 1/\epsilon + \log k - 3$. ■

5 The Statistical Setting

In this section we prove tight bounds for statistically-secure out-of-band authentication protocols in the group setting. First, in Section 5.1 we present our statistically-secure protocol. Then, in Section 5.2 we prove the security of our protocol, and in Section 5.3 we prove a matching lower bound on the length of the out-of-band authenticated value in any statistically-secure protocol.

5.1 Our Protocol

Our protocol, denoted π_{Stat}, is parametrized by the maximal forgery probability $\epsilon \in (0,1)$, integers $n, k \in \mathbb{N}$ denoting the length of each message and the number of receivers, respectively, and an odd integer $r \in \mathbb{N}$ denoting the number of rounds (we refer the reader to Section 1.2 for an intuitive overview of the protocol).

Notation. Denote the Galois field with q elements by $GF(q)$. Then, a message m of length n can be parsed as a polynomial of degree at most $\lceil n/\log q \rceil$ over $GF(q)$. Namely, a message $m = m_1, \ldots, m_t \in GF(q)^t$ defines a polynomial in the following manner: For every $x \in GF(q)$, we let $m(x) = \sum_{i=1}^{t} m_i \cdot x^i$. Then, for two distinct messages $m, \widehat{m} \in GF(q)^t$ and any two field elements $y, \widehat{y} \in GF(q)$, it holds that the polynomials $m(\cdot) + y$ and $\widehat{m}(\cdot) + \widehat{y}$ are distinct and thus $\Pr_{x \leftarrow GF(q)}\left[m(x) + y = \widehat{m}(x) + \widehat{y}\right] \leq t/q$. Let $\epsilon' = \epsilon/k$, and let $n_1 = n$. For every $j \in [r-1]$ let q_j be a prime number chosen in a deterministically and agreed upon manner in the interval $\left[\frac{2^{r-j} \cdot n_j}{\epsilon'}, \frac{2^{r-j+1} \cdot n_j}{\epsilon'}\right)$, and let $n_{j+1} = \lceil 2 \log q_j \rceil$.

Our protocol π_{Stat} is then defined by the following steps:

1. For every $i \in [k]$, S sends $m_{S,i}^1 = m_i$ to R_i. Denote by $m_{R_i}^1$ the string received by R_i.
2. For $j = 1$ to $r - 2$:
   (a) If j is odd, then for every $i \in [k]$:
      i. S chooses $y_i^j \leftarrow GF(q_j)$ and sends it to R_i.
      ii. R_i receives $\widehat{y_i^j}$, chooses $x_i^j \leftarrow GF(q_j)$ and sends it to S.
      iii. S receives $\widehat{x_i^j}$ and computes $m_{S,i}^{j+1} = \widehat{x_i^j} \| m_{S,i}^j(\widehat{x_i^j}) + y_i^j$.
      iv. R_i computes $m_{R_i}^{j+1} = x_i^j \| m_{R_i}^j(x_i^j) + \widehat{y_i^j}$.
   (b) if j is even, then for every $i \in [k]$:
      i. R_i chooses $y_i^j \leftarrow GF(q_j)$ and sends it to S.
      ii. S receives $\widehat{y_i^j}$, chooses $x_i^j \leftarrow GF(q_j)$ and sends it to R_i.
      iii. R_i receives $\widehat{x_i^j}$ and computes $m_{R_i}^{j+1} = \widehat{x_i^j} \| m_{R_i}^j(\widehat{x_i^j}) + y_i^j$.
      iv. S computes $m_{S,i}^{j+1} = x_i^j \| m_{S,i}^j(x_i^j) + \widehat{y_i^j}$.
3. For every $i \in [k]$, R_i chooses $y_i^{r-1} \leftarrow GF(q_j)$ and sends it to S.
4. S receives $\widehat{y_1^{r-1}}, \ldots, \widehat{y_k^{r-1}}$, chooses $x^{r-1} \leftarrow GF(q_{r-1})$, and for every $i \in [k]$ sends $x_i^{r-1} = x^{r-1}$ to R_i.
5. For every $i \in [k]$, R_i receives $\widehat{x_i^{r-1}}$ and computes $\widehat{\sigma_i} = m_{R_i}^{r-1}(\widehat{x_i^{r-1}}) + y_i^{r-1}$. Denote $m_{R_i}^r = \widehat{x_i^{r-1}} \| \widehat{\sigma_i}$.
6. For every $i \in [k]$, S computes $\sigma_i = m_{S,i}^{r-1}(x^{r-1}) + \widehat{y_i^{r-1}}$. Denote $m_{S,i}^r = x^{r-1} \| \sigma_i$. S sends $x^{r-1} \| \sigma_1 \| \ldots \| \sigma_k$ over the authenticated channel.
7. For every $i \in [k]$, if $m_{S,i}^r = m_{R_i}^r$ (i.e., if $x^{r-1} = \widehat{x_i^{r-1}}$ and $\sigma_i = \widehat{\sigma_i}$), R_i outputs $m_{R_i}^1$. Otherwise, R_i outputs $\perp$.

The following theorem (when the protocol is invoked with at least $\log^* n$ rounds) implies Theorem 1.3 as an immediate corollary:

Theorem 5.1. *Let $n, k \in \mathbb{N}$, let $r \geq 3$, and let $\epsilon \in (0,1)$. Then, protocol π_{Stat} is a statistically-secure out-of-band $(n, \ell, k, r, \epsilon)$-authentication protocol, where $\ell = (k+1) \cdot \left(\log \frac{1}{\epsilon} + \log k + \log^{(r-1)} n + O(1)\right)$.*

The correctness of our protocol is straightforward. In Lemma 5.2 we bound the length ℓ of the out-of-band authenticated value as stated in Theorem 5.1, and the proof of unforgeability is given in Section 5.2, yielding Theorem 5.1. A corollary of Lemma 5.2 is that when invoked with $r = \Omega(\log^* n)$, the sender in protocol π_{Stat} has to authenticate at most $(k+1) \cdot (\log(1/\epsilon) + \log k + O(1))$ bits.

Lemma 5.2. *Let $n, k \in \mathbb{N}$, let $r \geq 3$, and let $\epsilon \in (0,1)$. Then, in protocol π_{Stat} it holds that $\ell \leq (k+1) \cdot \left(\log \frac{1}{\epsilon} + \log k + \log^{(r-1)} n + O(1)\right)$.*

The proof of Lemma 5.2 will make use of the following two claims.

Claim 5.3. *If $n_j > 2^{r-j}/\epsilon'$ for every $j \in [r-2]$, then $n_{j+1} \leq \max\{4 \log^{(j)} n + 4 \log 5 + 3, 27\}$ for every $j \in [k-2]$.*

Proof. The proof is by induction on j. Since $n_j > 2^{r-j}/\epsilon'$ for every $j \in [r-2]$, it holds that for every $j \in [r-2]$,

$$q_j < \frac{2^{r-j+1}}{\epsilon'} \cdot n_j \leq 2n_j^2.$$

This implies that for every $j \in [r-2]$, it holds that

$$n_{j+1} = \lceil 2 \log q_j \rceil < \lceil 2 \log \left(2n_j^2\right) \rceil \leq 4 \log n_j + 3.$$

For $j = 1$, the claim indeed yields: $n_2 < 4 \log n + 3$. For $2 \leq j \leq r-2$, if $n_j \leq 27$, then $n_{j+1} < 4 \log 27 + 3 < 23$. Otherwise, by the induction hypothesis, it holds that

$$n_{j+1} \leq 4 \log n_j + 3 \leq 4 \log \left(4 \log^{(j-1)} n + 4 \log 5 + 3\right) + 3.$$

Consider the following two cases:

1. If $\log^{(j-1)} n \leq 4 \log 5 + 3$, then $n_{j+1} \leq 4 \log(20 \log 5 + 15) + 3 < 27$.
2. If $\log^{(j-1)} n > 4 \log 5 + 3$, then $n_{j+1} \leq 4 \log \left(5 \log^{(j-1)} n\right) + 3 = 4 \log^{(j)} n + 4 \log 5 + 3$.

■

Claim 5.4. *If $n_j \leq 2^{r-j}/\epsilon'$ for some $j \in [r-2]$, then for every $j' \in \{j, \ldots, r-2\}$, it holds that $n_{j'} \leq 2^{r-j'}/\epsilon'$.*

Proof. Assume $n_j \leq 2^{r-j}/\epsilon'$ for some $j \in [r-3]$. We prove $n_{j+1} \leq 2^{r-j-1}/\epsilon'$ and the claim follows.

By the assumption on n_j, it holds that

$$\begin{aligned} n_{j+1} &= \lceil 2 \log q_j \rceil \\ &\le \left\lceil 2 \log \left(\frac{2^{r-j}}{\epsilon'} \cdot n_j \right) \right\rceil \\ &\le \left\lceil 4 \log \left(\frac{2^{r-j}}{\epsilon'} \right) \right\rceil \\ &\le 4 \cdot \left(r - j + \log \frac{1}{\epsilon'} \right) + 1 \\ &\le 2^{r-j+\log \frac{1}{\epsilon'} - 1} \\ &= \frac{2^{r-j-1}}{\epsilon'}. \end{aligned}$$

The last inequality follows by the fact that $4x+1 \le 2^{x-1}$ for any $x \ge 6$ (if $r - j + \log(1/\epsilon') < 6$ then the parties can jump to Step 3 of the protocol and complete it, while S only has to send $(k+1) \cdot O(1)$ bits over the out-of-band channel, which implies Lemma 5.2). ■

We are now ready to prove Lemma 5.2.

Proof of Lemma 5.2. Informally speaking, we prove that q_{r-1} is at most roughly $1/\epsilon'$, and then the lemma follows, since S authenticates to $k+1$ elements in $GF(q_{r-1})$, which can be encoded using $\lceil (k+1) \cdot \log q_{r-1} \rceil$ bits.

More formally, we consider two separate cases. First we consider the case where $n_j > 2^{r-j}/\epsilon'$ for every $j \in [r-2]$. By Claim 5.3, it holds that $n_{r-1} \le \max \left\{ 4 \log^{(r-2)} n + 4 \log 5 + 3, 27 \right\}$. If $n_{r-1} \le 27$, then $q_{r-1} < 4 \cdot 27/\epsilon'$, and then

$$\begin{aligned} \ell &= \lceil (k+1) \cdot \log q_{r-1} \rceil \\ &\le (k+1) \cdot \left(\log \frac{1}{\epsilon'} + O(1) \right) \\ &= (k+1) \cdot \left(\log \frac{1}{\epsilon} + \log k + O(1) \right). \end{aligned}$$

Otherwise, it holds that $n_{r-1} \le 4 \log^{(r-2)} n + 4 \log 5 + 3$. Hence,

$$\begin{aligned} \ell &= \lceil (k+1) \cdot \log q_{r-1} \rceil \\ &= \left\lceil (k+1) \cdot \log \left(\frac{4}{\epsilon'} \cdot n_{r-1} \right) \right\rceil \\ &\le (k+1) \cdot \left(\log \frac{1}{\epsilon} + \log k + \log^{(r-1)} n + O(1) \right). \end{aligned}$$

We now turn to consider the case where there exists some $j \in [r-2]$ such that $n_j \le 2^{r-j}/\epsilon'$. By Claim 5.4, this means that $n_{r-2} \le 4/\epsilon'$. Therefore,

$$n_{r-1} = \lceil 2 \log q_{r-2} \rceil \le \left\lceil 2 \log \frac{2^3}{\epsilon'} \cdot n_{r-2} \right\rceil \le 4 \log \frac{1}{\epsilon'} + 11.$$

Where this is the case, the parties can set $q_{r-1} = \Theta(1/\epsilon')$, and the security of the protocol is preserved. This is due to the fact that our proof of security (see Section 5.2) only relies on the fact that two distinct polynomials over $GF(q_{r-1})$ defined by n_{r-1}-bit strings evaluate to the same value on at most $\epsilon'/2$ field elements; i.e., ${q_{r-1}}^{-1} \cdot \lceil n_{r-1}/\log(1/\epsilon') \rceil \le \epsilon'/2$. If $q_{r-1} = \Theta(1/\epsilon')$, then indeed

$$\ell \le (k+1) \cdot \left(\log \frac{1}{\epsilon} + \log k + \log^{(r-1)} n + O(1) \right),$$

concluding the proof. ■

5.2 Proof of Security

In this section, we prove the unforgeability of our protocol π_{Stat}, proving Theorem 5.1. For an adversary A, let $\mathsf{Forge}_{A,i}$ denote the event in which R_i outputs $\widehat{m_i} \notin \{m_i, \bot\}$ in an execution of π_{Stat} with A, and let $\mathsf{Forge}_A = \bigcup_{i \in [k]} \mathsf{Forge}_{A,i}$. The following Lemma captures the unforgeability of π_{Stat}.

Lemma 5.5. *For any computationally unbounded adversary A, it holds that* $\Pr\left[\mathsf{Forge}_A\right] \le \epsilon$.

Proof. We prove that for every $i \in [k]$, any computationally unbounded adversary A succeeds in making R_i output a fraudulent message with probability at most $\epsilon' = \epsilon/k$ and the theorem thus follows by union bound. Note that if A fools R_i this in particular means that $m^1_{S,i} \ne m^1_{R_i}$ but $m^r_{S,i} = m^r_{R_i}$. Hence, there exists a round $j \in [r-1]$ such that $m^j_{S,i} \ne m^j_{R_i}$ but $m^{j+1}_{S,i} = m^{j+1}_{R_i}$; denote this event by Coll^j_i. We will prove that for every j, $\Pr\left[\mathsf{Coll}^j_i\right] \le \epsilon'/2^{r-j}$, and then by taking a union bound over all rounds, the probability of $\mathsf{Forge}_{A,i}$ is at most $\sum_{j=1}^{r-1} \Pr\left[\mathsf{Coll}^j_i\right] \le \sum_{j=1}^{r-1} \epsilon'/2^{r-j} < \epsilon'$.

Similarly to Section 4.2, we denote by $T(v)$ the time in which a message v in the protocol is sent and fixed. We analyze separately the case where the round index j is odd, and the case that it is even. We start by bounding $\Pr\left[\mathsf{Coll}^j_i\right]$ in case j is odd (R_i picks the evaluation point of the polynomial and S chooses the shift), and consider three possible attack timings:

1. $T(\widehat{x^j_i}) < T(x^j_i)$: In this case, R_i chooses x^j_i at random from the field only after $\widehat{x^j_i}$ was fixed and sent to S. Recall that $\widehat{x^j_i}$ is the first part of $m^{j+1}_{S,i}$ and x^j_i is the first part of $m^{j+1}_{R_i}$. Hence,
$$\Pr\left[\mathsf{Coll}^j_i\right] \le \Pr_{x^j_i \leftarrow GF(q_j)}\left[x^j_i = \widehat{x^j_i}\right] = \frac{1}{q_j} \le \frac{\epsilon'}{2^{r-j}}.$$

2. $T(\widehat{x^j_i}) \ge T(x^j_i)$ and $T(\widehat{y^j_i}) \ge T(y^j_i)$: In this case, if the adversary chooses $\widehat{x^j_i} \ne x^j_i$, then $\Pr\left[\mathsf{Coll}^j_i\right] = \Pr\left[m^{j+1}_{S,i} = m^{j+1}_{R_i}\right] = 0$. So for the remainder of the analysis of this case, we assume $\widehat{x^j_i} = x^j_i$. Since j is odd, it is always the case that $T(x^j_i) > T(\widehat{y^j_i})$; i.e., R_i chooses x^j_i after receiving $\widehat{y^j_i}$. Since we are also in the case where $T(\widehat{y^j_i}) \ge T(y^j_i)$, this means that R_i chooses x^j_i when $m^j_{S,i}, m^j_{R_i}, y^j_i$ and $\widehat{y^j_i}$ are all fixed. In particular, if $m^j_{S,i} \ne m^j_{R_i}$, then the polynomials $m^j_{S,i}(\cdot) + y^j_i$ and $m^j_{R_i}(\cdot) + \widehat{y^j_i}$ are two distinct polynomials of degree at most $\lceil n_j/\log q_j \rceil$. Hence,
$$\begin{aligned}\Pr\left[\mathsf{Coll}^j_i\right] &= \Pr_{x^j_i \leftarrow GF(q_j)}\left[m^j_{S,i} \ne m^j_{R_i} \wedge m^j_{S,i}(x^j_i) + y^j_i = m^j_{R_i}(x^j_i) + \widehat{y^j_i}\right] \\ &\le \frac{1}{q_j} \cdot \left\lceil \frac{n_j}{\log q_j} \right\rceil \le \frac{\epsilon'}{2^{r-j}}.\end{aligned}$$

3. $T(\widehat{x_i^j}) \geq T(x_i^j)$ and $T(\widehat{y_i^j}) < T(y_i^j)$: As before, if $\widehat{x_i^j} \neq x_i^j$, then $\Pr\left[\mathsf{Coll}_i^j\right] = 0$, so we assume $\widehat{x_i^j} = x_i^j$. In this case, S chooses y_i^j and R_i chooses x_i^j when the adversary has already chosen $\widehat{y_i^j}$. Since y_i^j and x_i^j are chosen independently, we may assume without loss of generality that $T(y_i^j) > T(x_i^j)$, meaning y_i^j is chosen when $m_{S,i}^j, m_{R_i}^j, \widehat{y_i^j}$ and x_i^j are already fixed (and thus also $\widehat{x_i^j}$, since we assume $\widehat{x_i^j} = x_i^j$). It follows that
$$\Pr\left[\mathsf{Coll}_i^j\right] = \Pr_{y_i^j \leftarrow GF(q_j)}\left[y_i^j = m_{R_i}^j(x_i^j) + \widehat{y_i^j} - m_{S,i}^j(x_i^j)\right] \leq \frac{1}{q_j} \leq \frac{\epsilon'}{2^{r-j}}.$$

We now turn to bound $\Pr\left[\mathsf{Coll}_i^j\right]$ in case j is even (S picks the evaluation point of the polynomial and R_i chooses the shift). The proof is very similar to the case where j is odd, and considers the same three cases:

1. $T(\widehat{x_i^j}) < T(x_i^j)$: In this case, S chooses x_i^j at random when $\widehat{x_i^j}$ is fixed. Therefore,
$$\Pr\left[\mathsf{Coll}_i^j\right] \leq \Pr_{x_i^j \leftarrow GF(q_j)}\left[x_i^j = \widehat{x_i^j}\right] = \frac{1}{q_j} \leq \frac{\epsilon'}{2^{r-j}}.$$

2. $T(\widehat{x_i^j}) \geq T(x_i^j)$ and $T(\widehat{y_i^j}) \geq T(y_i^j)$: As in the analysis for odd values of j, we can assume $\widehat{x_i^j} = x_i^j$, and we know that S chooses x_i^j when $m_{S,i}^j, m_{R_i}^j, y_i^j$ and $\widehat{y_i^j}$ are all fixed (in the last round, this follows also by the fact that S chooses x^{r-1} after receiving all $\widehat{y_i^{r-1}}$'s). In particular, if $m_{S,i}^j \neq m_{R_i}^j$, then the polynomials $m_{S,i}^j(\cdot) + \widehat{y_i^j}$ and $m_{R_i}^j(\cdot) + y_i^j$ are two distinct polynomials of degree at most $\lceil n_j / \log q_j \rceil$. Hence,
$$\Pr\left[\mathsf{Coll}_i^j\right] = \Pr_{x_i^j}\left[m_{S,i}^j \neq m_{R_i}^j \wedge m_{S,i}^j(x_i^j) + \widehat{y_i^j} = m_{R_i}^j(x_i^j) + y_i^j\right] \leq \frac{\epsilon'}{2^{r-j}}.$$

3. $T(\widehat{x_i^j}) \geq T(x_i^j)$ and $T(\widehat{y_i^j}) < T(y_i^j)$: As before, we assume $\widehat{x_i^j} = x_i^j$, and we know that R_i chooses y_i^j and S chooses x_i^j when the adversary has already chosen $\widehat{y_i^j}$. Since y_i^j and x_i^j are chosen independently, we may assume without loss of generality that $T(y_i^j) > T(x_i^j)$, meaning y_i^j is chosen when $m_{S,i}^j, m_{R_i}^j, \widehat{y_i^j}$ and x_i^j are already fixed. Hence,
$$\Pr\left[\mathsf{Coll}_i^j\right] = \Pr_{y_i^j \leftarrow GF(q_j)}\left[y_i^j = m_{S,i}^j(x_i^j) + \widehat{y_i^j} - m_{R_i}^j(x_i^j)\right] \leq \frac{\epsilon'}{2^{r-j}}.$$

Let $\mathsf{Coll}_i = \bigcup_{j \in [r-1]} \mathsf{Coll}_i^j$. By taking a union bound over all rounds, it follows that for every $i \in [k]$,
$$\Pr\left[\mathsf{Coll}_i\right] \leq \sum_{j=1}^{r-1} \Pr\left[\mathsf{Coll}_i^j\right] \leq \sum_{j=1}^{r-1} \frac{\epsilon'}{2^{r-j}} < \epsilon'.$$
Since for every $i \in [k]$, it is the case that $\mathsf{Forge}_{A,i}$ implies Coll_i, it holds that for every $i \in [k]$, $\Pr\left[\mathsf{Forge}_{A,i}\right] \leq \Pr\left[\mathsf{Coll}_i\right] \leq \epsilon'$. by taking a union bound over all receivers it holds that $\Pr\left[\mathsf{Forge}_A\right] \leq k \cdot \epsilon' = \epsilon$. ■

5.3 Lower Bound

In this section we present a lower bound on the number of bits the sender has to out-of-band authenticate in the group setting. We prove the following theorem:

Theorem 5.6. *For any statistically-secure out-of-band $(n, \ell, k, r, \epsilon)$-authentication protocol, if $n \geq (k+2) \cdot \log(1/\epsilon)$ then $\ell \geq (k+1) \cdot \log(1/\epsilon) - k$.*

Proof. Let $\pi = (S, R_1, \ldots, R_k)$ be a statistically-secure out-of-band $(n, \ell, k, r, \epsilon)$-authentication protocol. We assume without loss of generality that $r \equiv 1 \mod (k+1)$ and that π has the following structure. For every $j \in [r-1]$, in round j there exists a single "active" party that sends a message (over the insecure channels) to each of the other parties, and all other parties do not send any messages in that round. If $j \equiv 1 \mod (k+1)$, then the sender S is the active party in round j. Otherwise, if $j \equiv i+1 \mod (k+1)$ for some $i \in [k]$, then receiver R_i is the active user in round j. Denote the vector of messages sent in round j by x_{j-1} and the random variable describing that vector by X_{j-1} (so the vectors of messages sent over the insecure channels are $x_0, \ldots, x_{r-2}$). Finally, in round r, the sender S sends the short out-of-band authenticated value σ, and we denote the random variable describing it by Σ. We also denote the random variable describing the vector of input messages to S by M.

Observe, that we can write the Shannon entropy of Σ as

$$\begin{aligned}
\mathrm{H}(\Sigma) &= \mathrm{H}(\Sigma) - \mathrm{H}(\Sigma|M, X_0) + \sum_{j \in [r-2]} (\mathrm{H}(\Sigma|M, X_0, \ldots, X_{j-1}) - \mathrm{H}(\Sigma|M, X_0, \ldots, X_j)) \\
&\quad + \mathrm{H}(\Sigma|M, X_0, \ldots, X_{r-2}) \\
&= \mathrm{I}(\Sigma; M, X_0) + \sum_{j \in [r-2]} I(\Sigma; X_j|M, X_0, \ldots, X_{j-1}) + \mathrm{H}(\Sigma|M, X_0, \ldots, X_{r-2}) \\
&= \mathrm{I}(\Sigma; M, X_0) + \sum_{i \in \{0,\ldots,k\}} \sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} I(\Sigma; X_j|M, X_0, \ldots, X_{j-1}) + \mathrm{H}(\Sigma|M, X_0, \ldots, X_{r-2}).
\end{aligned}$$

To bound the above expression, we make use of the following two lemmata, which we will prove below. Intuitively speaking, Lemma 5.7 shows that the messages of the sender S during the execution of π need to reduce, on average, roughly $\log(1/\epsilon)$ bits of entropy from the out-of-band authenticated value.

Lemma 5.7. *If $n \geq 1/k \cdot \log(1/\epsilon)$, then*

$$\mathrm{I}(\Sigma; M, X_0) + \sum_{\substack{j \in [r-2]: \\ j \equiv 0 \mod (k+1)}} \mathrm{I}(\Sigma; X_j|M, X_0, \ldots, X_{j-1}) + \mathrm{H}(\Sigma|M, X_0, \ldots, X_{r-2}) \geq \log(1/\epsilon) - 1.$$

In a similar fashion, Lemma 5.8 shows that for any $i \in [k]$, the messages of receiver R_i during the execution of π need to reduce, on average, roughly $\log(1/\epsilon)$ bits of entropy from the out-of-band authenticated value.

Lemma 5.8. *If $n \geq (k+2) \cdot \log(1/\epsilon)$ and $\ell \leq (k+1) \cdot \log(1/\epsilon)$, then for every $i \in [k]$,*

$$\sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \mathrm{I}(\Sigma, X_j|M, X_0, \ldots, X_{j-1}) \geq \log(1/\epsilon) - 1.$$

Now, if $\ell > (k+1) \cdot \log(1/\epsilon)$, then the theorem follows. Otherwise, by Lemmata 5.7 and 5.8 it holds that, $\ell \geq \mathrm{H}(\Sigma) \geq (k+1) \cdot \log(1/\epsilon) - k$, concluding the proof of Theorem 5.6. ■

We now turn to prove the two lemmata used in the proof of Theorem 5.6, starting with the proof of Lemma 5.7.

Proof of Lemma 5.7. Consider a computationally unbounded adversary A that launches the following attack:

1. A picks k random messages $\widehat{\mathbf{m}} = (\widehat{m_1}, \ldots, \widehat{m_k}) \leftarrow \{0,1\}^{n \times k}$ and runs an honest execution of π with the receivers (where A simulates S). Denote the authentication string sent in this execution by $\widehat{\sigma}$.

2. A chooses $\mathbf{m} = (m_1, \ldots, m_k) \leftarrow \{0,1\}^{n \times k}$ as the input to S, and starts an execution of π with S. Namely, A gets x_0 from S.

3. If $\Pr[\mathbf{m}, x_0, \widehat{\sigma}] = 0$, A quits. Otherwise, it samples $(\widehat{x_1}, \ldots, \widehat{x_k})$ from the conditional distribution $(X_1, \ldots, X_k) | (\mathbf{m}, x_0, \widehat{\sigma})$, and sends the corresponding messages to S, who replies with the messages x_{k+1}. The attack then continues in a similar manner. After each message vector x_j sent by S, A checks if $\Pr[\mathbf{m}, x_0, \widehat{x_1}, \ldots, x_j, \widehat{\sigma}] = 0$. If so, A quits. Otherwise, A samples $(\widehat{x_{j+1}}, \ldots, \widehat{x_{j+k}})$ from the distribution $(X_{j+1}, \ldots, X_{j+k}) | (\mathbf{m}, x_0, \widehat{x_1}, \ldots, x_j, \widehat{\sigma})$ and sends the corresponding messages of rounds $j+2, \ldots, j+k+1$ to S.

4. In round r, S sends an authentication string σ over the out-of-band channel. If $\sigma \neq \widehat{\sigma}$, A quits. Otherwise, A forwards σ to the receivers.

We first wish to bound the probability that some receiver does not output $\perp$. For ease of notation, we denote by Acc the event in which every receiver R_i outputs $\widehat{m_i}$ in the above attack. For every $j \in [r-2]$, we also denote by t_j the transcript of the execution with S until (and including) the messages x_j of round $j+1$. Then, since the x_j's are chosen conditioned on $\mathbf{m}$ and t_{j-1}, and the tuples $(\widehat{x_{j+1}}, \ldots, \widehat{x_{j+k}})$ are chosen conditioned in $\mathbf{m}, t_{j-1}$ and $\widehat{\sigma}$, it holds that

$$\begin{aligned}
\Pr[\mathsf{Acc}] &= \sum_{\widehat{\sigma} \in \{0,1\}^{\ell}} \Pr[\widehat{\sigma}] \cdot \Pr[\mathsf{Acc} | \widehat{\sigma}] \\
&= \sum_{\widehat{\sigma}} \Pr[\widehat{\sigma}] \cdot \sum_{\mathbf{m}, x_0} \Pr[\mathbf{m}, x_0] \cdot \Pr[\mathsf{Acc} | \mathbf{m}, x_0, \widehat{\sigma}] \\
&= \sum_{\substack{\mathbf{m}, x_0, \widehat{\sigma}: \\ \Pr[\mathbf{m}, x_0, \widehat{\sigma}] > 0}} \Pr[\widehat{\sigma}] \cdot \Pr[\mathbf{m}, x_0] \cdot \sum_{\widehat{x_1}, \ldots, \widehat{x_k}} \Pr[\widehat{x_1}, \ldots, \widehat{x_k} | \mathbf{m}, x_0, \widehat{\sigma}] \\
&\qquad \cdot \Pr[\mathsf{Acc} | \mathbf{m}, t_k, \widehat{\sigma}] \\
&= \sum_{\substack{\mathbf{m}, t_k, \widehat{\sigma}: \\ \Pr[\mathbf{m}, t_k, \widehat{\sigma}] > 0}} \Pr[\widehat{\sigma}] \cdot \Pr[\mathbf{m}, x_0] \cdot \Pr[\widehat{x_1}, \ldots, \widehat{x_k} | \mathbf{m}, x_0, \widehat{\sigma}] \\
&\qquad \cdot \sum_{x_{k+1}} \Pr[x_{k+1} | \mathbf{m}, t_k] \cdot \Pr[\mathsf{Acc} | \mathbf{m}, t_k, \widehat{\sigma}]
\end{aligned}$$

$$\begin{aligned}
= \sum_{\substack{\mathbf{m},t_{r-2},\widehat{\sigma}: \\ \Pr[\mathbf{m},t_{r-2},\widehat{\sigma}]>0}} & \Pr\left[\widehat{\sigma}\right]\cdot\Pr\left[\mathbf{m},x_0\right] \\
& \cdot \prod_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} \Pr\left[\widehat{x_{j-k}},\dots,\widehat{x_{j-1}}|\mathbf{m},t_{j-k-1},\widehat{\sigma}\right]\cdot\Pr\left[x_j|\mathbf{m},t_{j-1}\right] \\
& \cdot \Pr\left[\widehat{x_{r-k-1}},\dots,\widehat{x_{r-2}}|\mathbf{m},t_{r-k-2},\widehat{\sigma}\right]\cdot\Pr\left[\mathsf{Acc}|\mathbf{m},t_{r-2},\widehat{\sigma}\right].
\end{aligned}$$

By the correctness of π:

$$\begin{aligned}
\Pr\left[\mathsf{Acc}\right] \geq \sum_{\substack{\mathbf{m},t_{r-2},\widehat{\sigma}: \\ \Pr[\mathbf{m},t_{r-2},\widehat{\sigma}]>0}} & \Pr\left[\widehat{\sigma}\right]\cdot\Pr\left[\mathbf{m},x_0\right] \\
& \cdot \prod_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} \Pr\left[\widehat{x_{j-k}},\dots,\widehat{x_{j-1}}|\mathbf{m},t_{j-k-1},\widehat{\sigma}\right]\cdot\Pr\left[x_j|\mathbf{m},t_{j-1}\right] \\
& \cdot \Pr\left[\widehat{x_{r-k-1}},\dots,\widehat{x_{r-2}}|\mathbf{m},t_{r-k-2},\widehat{\sigma}\right]\cdot\Pr\left[\widehat{\sigma}|\mathbf{m},t_{r-2}\right] \\
= \sum_{\substack{\mathbf{m},t_{r-2},\widehat{\sigma}: \\ \Pr[\mathbf{m},t_{r-2},\widehat{\sigma}]>0}} & \Pr\left[\mathbf{m},x_0,\widehat{\sigma}\right]\cdot\frac{\Pr\left[\widehat{\sigma}\right]}{\Pr\left[\widehat{\sigma}|\mathbf{m},x_0\right]} \\
& \cdot \prod_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} \Pr\left[\widehat{x_{j-k}},\dots,\widehat{x_{j-1}},x_j|\mathbf{m},t_{j-k-1},\widehat{\sigma}\right]\cdot\frac{\Pr\left[x_j|\mathbf{m},t_{j-1}\right]}{\Pr\left[x_j|\mathbf{m},t_{j-1},\widehat{\sigma}\right]} \\
& \cdot \Pr\left[\widehat{x_{r-k-1}},\dots,\widehat{x_{r-2}}|\mathbf{m},t_{r-k-2},\widehat{\sigma}\right]\cdot\Pr\left[\widehat{\sigma}|\mathbf{m},t_{r-2}\right] \\
= \sum_{\substack{\mathbf{m},t_{r-2},\widehat{\sigma}: \\ \Pr[\mathbf{m},t_{r-2},\widehat{\sigma}]>0}} & \Pr\left[\mathbf{m},t_{r-2},\widehat{\sigma}\right]\cdot\frac{\Pr\left[\widehat{\sigma}\right]}{\Pr\left[\widehat{\sigma}|\mathbf{m},x_0\right]} \\
& \cdot \left(\prod_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} \frac{\Pr\left[x_j|\mathbf{m},t_{j-1}\right]}{\Pr\left[x_j|\mathbf{m},t_{j-1},\widehat{\sigma}\right]}\right)\cdot\Pr\left[\widehat{\sigma}|\mathbf{m},t_{r-2}\right] \\
= \sum_{\substack{\mathbf{m},t_{r-2},\widehat{\sigma}: \\ \Pr[\mathbf{m},t_{r-2},\widehat{\sigma}]>0}} & \Pr\left[\mathbf{m},t_{r-2},\widehat{\sigma}\right] \\
& \cdot 2^{-\left(\log\frac{\Pr[\widehat{\sigma}|\mathbf{m},x_0]}{\Pr[\widehat{\sigma}]}+\left(\sum_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}}\log\frac{\Pr\left[x_j|\mathbf{m},t_{j-1},\widehat{\sigma}\right]}{\Pr\left[x_j|\mathbf{m},t_{j-1}\right]}\right)+\log\frac{1}{\Pr[\widehat{\sigma}|\mathbf{m},t_{r-2}]}\right)}.
\end{aligned}$$

By Jensen's inequality, it holds that

$$\Pr\left[\mathsf{Acc}\right] \geq 2^{-\left(\mathrm{I}(\Sigma;M,X_0)+\sum_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}}\mathrm{I}(\Sigma;X_j|M,X_0,\dots,X_{j-1})+\mathrm{H}(\Sigma|M,X_0,\dots,X_{r-2})\right)}.$$

Finally, denote by Forge_A the event in which A succeeds in making one of the receivers output a fraudulent message. Then, as in Section 4.3, it holds that

$$\begin{aligned}\Pr\left[\mathsf{Forge}_A\right] &\geq \Pr\left[\mathsf{Acc} \wedge \exists i \in [k] \text{ s.t. } \widehat{m_i} \neq m_i\right] \\ &\geq \Pr\left[\mathsf{Acc}\right] - \Pr\left[\forall i \in [k] : \widehat{m_i} = m_i\right] = \Pr\left[\mathsf{Acc}\right] - 2^{k \cdot n}.\end{aligned}$$

By the unforgeability of the protocol π and by the assumption that $n \geq 1/k \cdot \log(1/\epsilon)$, the above yields $2\epsilon \geq \Pr\left[\mathsf{Acc}\right]$. It thus holds that

$$2\epsilon \geq 2^{-\left(\mathrm{I}(\Sigma;M,X_0)+\sum_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} \mathrm{I}(\Sigma;X_j|M,X_0,\ldots,X_{j-1})+\mathrm{H}(\Sigma|M,X_0,\ldots,X_{r-2})\right)}.$$

Equivalently,

$$\begin{aligned}\mathrm{I}(\Sigma; M, X_0) + \sum_{\substack{j\in[r-2]: \\ j\equiv 0 \mod (k+1)}} &\mathrm{I}(\Sigma; X_j | M, X_0, \ldots, X_{j-1}) \\ &+\mathrm{H}(\Sigma | M, X_0, \ldots, X_{r-2}) \geq \log(1/\epsilon) - 1.\end{aligned}$$

■

The proof of Lemma 5.8 is quite similar in nature to that of Lemma 5.7.

Proof of Lemma 5.8. For every $i \in [k]$, consider an adversary A, defined by the following attack aimed at making receiver R_i output a fraudulent message:

1. A picks k random messages $\mathbf{m} = (m_1, \ldots, m_k) \leftarrow \{0,1\}^{n \times k}$ as the input to S and runs an honest execution of π with the S and with all receivers but R_i (where A simulates R_i). Denote the authentication string sent in this execution by σ. A reads σ and delays it from being delivered to R_i.

2. A starts an execution of π with R_i in the following manner. A samples $\widehat{\mathbf{m}} = (\widehat{m_1}, \ldots, \widehat{m_k})$ and messages $\widehat{x_0}, \ldots, \widehat{x_{i-1}}$, from the conditional distribution $(M, X_0, \ldots, X_{i-1})|\sigma$, sets $\widehat{\mathbf{m}}$ as the input to (the simulated) sender in the execution, and sends to R_i the relevant from $\widehat{x_0}, \ldots, \widehat{x_{i-1}}$ in rounds $1, \ldots, i$ of the execution. R_i replies with x_i.

3. If $\Pr\left[\widehat{\mathbf{m}}, \widehat{x_0}, \ldots, \widehat{x_{i-1}}, x_i, \widehat{\sigma}\right] = 0$, A quits. Otherwise, A samples a tuple $(\widehat{x_{i+1}}, \ldots, \widehat{x_{i+k}})$ from the conditional distribution $(X_{i+1}, \ldots, X_{i+k})|(\widehat{\mathbf{m}}, \widehat{x_0}, \ldots, \widehat{x_{i-1}}, x_i, \widehat{\sigma})$, and sends the corresponding messages to S, who replies with the messages x_{k+1+i}. The attack then continues in a similar manner. After each message vector x_j sent by R_i, A checks if $\Pr\left[\widehat{\mathbf{m}}, \widehat{x_0}, \ldots, \widehat{x_{j-1}}, x_j, \widehat{\sigma}\right] = 0$. If that is the case, A quites. Otherwise, A samples $(\widehat{x_{j+1}}, \ldots, \widehat{x_{j+k}})$ according to the distribution $(X_{j+1}, \ldots, X_{j+k})|(\widehat{\mathbf{m}}, \widehat{x_0}, \ldots, \widehat{x_{j-1}}, x_j, \widehat{\sigma})$ and sends the corresponding messages of rounds $j+2, \ldots, j+k+1$ to R_i, who replies with x_{j+k+1}.[14]

4. In round r, A forwards σ to the R_i.

We now denote by Acc the event in which R_i outputs $\widehat{m_i}$. We continue similarly to our analysis in the proof of Lemma 5.7: We note that the x_j's are chosen conditioned on t_{j-1}, $\widehat{\mathbf{m}}$ is chosen conditioned on σ and the $\widehat{x_j}$'s are chosen conditioned on $\widehat{\mathbf{m}}, t_{j-1}$ and σ. Moreover, conditioned on

[14] If $j + k \geq r - 2$, then A samples $(\widehat{x_{j+1}}, \ldots, \widehat{x_{r-2}})$, and R_i does not reply.

t_{j-1}, X_j is independent of the choice of $\widehat{\mathbf{m}}$, so for any x_j, t_{j-1} and $\widehat{\mathbf{m}}$, it holds that $\Pr[x_j|t_{j-1}] = \Pr[x_j|\widehat{\mathbf{m}}, t_{j-1}]$. Hence,

$$\begin{aligned}
\Pr[\mathsf{Acc}] \geq & \sum_{\substack{\widehat{\mathbf{m}}, t_{r-2}, \sigma: \\ \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] > 0}} \Pr[\widehat{\mathbf{m}}, \widehat{x_0}, \ldots, \widehat{x_{i-1}}, \sigma] \cdot \Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}] \\
& \cdot \prod_{\substack{j \in [r-k-3]: \\ j \equiv i \mod (k+1)}} \Pr\left[\widehat{x_{j+1}}, \ldots, \widehat{x_{j+k}} | \widehat{\mathbf{m}}, t_j, \sigma\right] \cdot \Pr[x_{j+k+1}|\widehat{\mathbf{m}}, t_{j+k}] \\
& \cdot \Pr\left[\widehat{x_{r-k-1+i}}, \ldots, \widehat{x_{r-2}} | \widehat{\mathbf{m}}, t_{r-k-2+i}, \sigma\right] \text{[15]} \\
= & \sum_{\substack{\widehat{\mathbf{m}}, t_{r-2}, \sigma: \\ \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] > 0}} \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] \cdot \prod_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \frac{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}]}{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}, \sigma]} \\
= & \sum_{\substack{\widehat{\mathbf{m}}, t_{r-2}, \sigma: \\ \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] > 0}} \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] \cdot 2^{-\left(\sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \log \frac{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}, \sigma]}{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}]}\right)}.
\end{aligned}$$

By Jensen's inequality,

$$\begin{aligned}
\Pr[\mathsf{Acc}] & \geq 2^{-\left(\sum_{\substack{\widehat{\mathbf{m}}, t_{r-2}, \sigma: \\ \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] > 0}} \Pr[\widehat{\mathbf{m}}, t_{r-2}, \sigma] \cdot \sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \log \frac{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}, \sigma]}{\Pr[x_i|\widehat{\mathbf{m}}, t_{i-1}]}\right)} \\
& = 2^{-\left(\sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \mathrm{I}(\Sigma, X_j | M, X_0, \ldots, X_{j-1})\right)}.
\end{aligned}$$

Similarly to Lemma 5.7, we denote by Forge_A the event in which A succeeds in making R_i output a fraudulent message. Then, it holds that

$$\Pr[\mathsf{Forge}_A] \geq \Pr[\mathsf{Acc} \wedge \widehat{m_i} \neq m_i] \geq \Pr[\mathsf{Acc}] - \Pr[\widehat{m_i} = m_i].$$

Unlike in Lemma 5.7, the input messages $\widehat{\mathbf{m}}$ – and in particular $\widehat{m_i}$ – are now chosen conditioned on σ. So, it holds that

$$\begin{aligned}
\Pr[\widehat{m_i} = m_i] & = \sum_{\sigma \in \{0,1\}^\ell} \Pr[\sigma] \cdot \sum_{m_i \in \{0,1\}^n} (\Pr[m_i|\sigma])^2 \\
& \leq \sum_\sigma \Pr[\sigma] \cdot \max_{m_i} \Pr[m_i|\sigma] \cdot \sum_{m_i} \Pr[m_i|\sigma] \\
& = \sum_\sigma \Pr[\sigma] \cdot \max_{m_i} \Pr[m_i|\sigma] \leq \sum_\sigma \max_{m_i} \Pr[m_i].
\end{aligned}$$

Since m_i is chosen uniformly at random from $\{0,1\}^n$, it follows that $\Pr[\widehat{m_i} = m_i] \leq 2^{-n+\ell}$. Taking into consideration the assumption that $\ell \leq (k+1) \cdot \log(1/\epsilon)$ and $n \geq (k+2) \cdot \log(1/\epsilon)$, it follows that $2 \cdot \Pr[\mathsf{Forge}_A] \leq \Pr[\mathsf{Acc}]$. Finally, the unforgeability of π yields

$$2^{-\left(\sum_{\substack{j \in [r-2]: \\ j \equiv i \mod (k+1)}} \mathrm{I}(\Sigma, X_j | M, X_0, \ldots, X_{j-1})\right)} \leq 2\epsilon,$$

[15] If $i = k$, then we set $\Pr\left[\widehat{x_{r-k-1+i}}, \ldots, \widehat{x_{r-2}} | \widehat{\mathbf{m}}, t_{r-k-2+i}, \sigma\right] = 1$.

or equivalently,

$$\sum_{\substack{j\in[r-2]:\\ j\equiv i \mod (k+1)}} \mathrm{I}(\Sigma, X_j|M, X_0, \ldots, X_{j-1}) \geq \log(1/\epsilon) - 1.$$

■

Acknowledgments

We thank Ilya Mironov and Moni Naor for very useful discussions in various stages of this work.

References

[BM94] S. M. Bellovin and M. Merritt. An attack on the Interlock protocol when used for authentication. *IEEE Transactions on Information Theory*, 40(1):273–275, 1994.

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

[BSJ+17] M. Bellare, A. C. Singh, J. Jaeger, M. Nyayapati, and I. Stepanovs. Ratcheted encryption and key exchange: The security of messaging. In *Advances in Cryptology – CRYPTO '17*, pages 619–650, 2017.

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

[CF01] R. Canetti and M. Fischlin. Universally composable commitments. In *Advances in Cryptology – CRPYTO '01*, pages 19–40, 2001.

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

[CIO98] G. D. Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 141–150, 1998.

[CKO+01] G. D. Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. Efficient and non-interactive non-malleable commitment. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 40–59, 2001.

[COS+17] M. Ciampi, R. Ostrovsky, L. Siniscalchi, and I. Visconti. Four-round concurrent non-malleable commitments from one-way functions. In *Advances in Cryptology – CRYPTO '17*, pages 127–157, 2017.

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

[DG03] I. Damgard and J. Groth. Non-interactive and reusable non-malleable commitment schemes. In *Proceedings of the 35th annual ACM Symposium on Theory of Computing*, pages 426–437, 2003.

[Ell96] C. M. Ellison. Establishing identity without certification authorities. In *Proceedings of the 6th USENIX Security Symposium*, pages 7–7, 1996.

[FF00] M. Fischlin and R. Fischlin. Efficient non-malleable commitment schemes. In *Advances in Cryptology – CRYPTO '00*, pages 413–431, 2000.

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

[Gol01] O. Goldreich. Foundations of Cryptography – Volume 1: Basic Techniques. Cambridge University Press, 2001.

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

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

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

[KBB17] N. Kobeissi, K. Bhargavan, and B. Blanchet. 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)*, pages 435–450, 2017.

[LP11] H. Lin and R. Pass. Constant-round non-malleable commitments from any one-way function. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pages 705–714, 2011.

[LPV08] H. Lin, R. Pass, and M. Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *Proceedings of the 5th Theory of Cryptography Conference*, pages 571–588, 2008.

[Nao91] M. Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[NSS06] M. Naor, G. Segev, and A. Smith. Tight bounds for unconditional authentication protocols in the manual channel and shared key models. In *Advances in Cryptology – CRYPTO'06*, pages 214–231, 2006.

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

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

[PR05] R. Pass and A. Rosen. Concurrent non-malleable commitments. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572, 2005.

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

[PV06] S. Pasini and S. Vaudenay. An optimal non-interactive message authentication protocol. In *Topics in Cryptology – CT-RSA '06*, pages 280–294, 2006.

[RMS18] P. Rösler, C. Mainka, and J. Schwenk. 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] R. L. Rivest and A. Shamir. How to expose an eavesdropper. *Communications of the ACM*, 27(4):393–395, 1984.

[Tela] Telegram. End-to-end encrypted voice calls – key verification. Available at `https://core.telegram.org/api/end-to-end/voice-calls#key-verification` (accessed 16-May-2018).

[Telb] Telegram. End-to-end encryption. Available at `https://core.telegram.org/api/end-to-end` (accessed 16-May-2018).

[Vau05] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology – CRYPTO'05*, pages 309–326, 2005.

[Vib] Viber encryption overview. Available at `https://www.viber.com/app/uploads/Viber-Encryption-Overview.pdf` (accessed 16-May-2018).

[Wha] WhatsApp encryption overview. Available at `https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf` (accessed 16-May-2018).

[Wik] Wikipedia. Instant messaging. Available at `https://en.wikipedia.org/wiki/Instant_messaging` (accessed 16-May-2018).

A A Concrete Example: Out-of-Band Authenticating a Group Diffie-Hellman Protocol

In this section we present a possible instantiation of our computationally-secure out-of-band authentication protocol (see Section 4.1) when used for authenticating (the transcript of) a group key-exchange protocol. The instantiation to be described below illustrates the practicality and efficiency of our protocol, when used to fulfill a very natural task.

Let $\mathcal{G}$ be a group of prime order q that is generated by $g \in \mathcal{G}$. Consider the following group key-exchange protocol for a group consisting of $k+1$ parties, denoted S and $R_1, \dots, R_k$:

1. For every $i \in [k]$, R_i chooses a random $b_i \leftarrow \mathbb{Z}_q$, computes $B_i = g^{b_i}$ and sends B_i to S.
2. S chooses a random $a \leftarrow \mathbb{Z}_q$, and for every $i \in [k]$ computes $D_i = B_i^a$, and sends D_i to R_i.
3. The joint group key is then $K = g^a$. Note that S knows a and can compute K directly, while each R_i knows b_i and can compute $K = D_i^{b_i^{-1}} = \left(g^{ab_i}\right)^{b_i^{-1}} = g^a$.

The security of this protocol in the presence of an eavesdropper follows easily from the decisional Diffie-Hellman (DDH) assumption. However, a man-in-the-middle adversary is capable of devastating attacks (exactly as in the two-party Diffie-Hellman protocol).

This is where our out-of-band authentication protocol comes into the picture: A group may employ our protocol on top of the above key-exchange protocol, in order to immune it from man-in-the-middle attacks. In the example below, we instantiate the commitment scheme used in our protocol via any off-the-shelf cryptographic hash function H (e.g., SHA-2). As noted in Section 4.2, security is guaranteed when H is modeled as a random oracle.

The protocol thus has the following form:

1. For every $i \in [k]$, R_i randomly chooses $b_i \leftarrow \mathbb{Z}_q$ and computes $B_i = g^{b_i}$. R_i also chooses $r_i \leftarrow \{0,1\}^\ell$ and $s_i \leftarrow \{0,1\}^\lambda$ uniformly at random and computes $c_i = H(r_i \| s_i)$. R_i sends (B_i, c_i) to S; R_i also sends c_i to all other receivers.[16] Denote the values received by S by $(\widehat{B_i}, \widehat{c_i})$ and the value received by receiver R_j by $\widehat{c_{i\to j}}$ for every $j \in [k] \setminus \{i\}$.

2. S chooses a random $a \leftarrow \mathbb{Z}_q$ and for every $i \in [k]$ computes $D_i = \widehat{B_i}^{\,a}$; denote $m_i = (\widehat{B_i}, \widehat{c_i}, D_i)$. S then chooses $r_S \leftarrow \{0,1\}^\ell$ and for every $i \in [k]$, she chooses a random $s^i_S \leftarrow \{0,1\}^\lambda$ and computes $c^i_S = H(m_i \| r_S \| s^i_S)$. S sends (D_i, c^i_S) to R_i. Denote the values received by R_i by $(\widehat{D_i}, \widehat{c^i_S})$ and let $\widehat{m_i} = (B_i, c_i, \widehat{D_i})$.

3. For every $i \in [k]$, R_i sends r_i and s_i to all other parties. Denote the values received by S by $(\widehat{r_i}, \widehat{s_i})$ and the values received by R_j by $(\widehat{r_{i\to j}}, \widehat{s_{i\to j}})$ for every $j \in [k] \setminus \{i\}$. S verifies that $\widehat{c_i} = H(\widehat{r_i} \| \widehat{s_i})$ and each R_j for every $j \in [k] \setminus \{i\}$ verifies that $\widehat{c_{i\to j}} = H(\widehat{r_{i\to j}}, \widehat{s_{i\to j}})$. If the verification by a certain party fails, that party outputs $\perp$ and terminates.

4. S sends (r_S, s^i_S) to receiver R_i, for every $i \in [k]$. Denote the values received by R_i by $(\widehat{r^i_S}, \widehat{s^i_S})$. Each R_i then verifies that $\widehat{c^i_S} = H(\widehat{m_i} \| \widehat{r^i_S} \| \widehat{s^i_S})$. If the verification by a certain party fails, that party outputs $\perp$ and terminates.

5. S computes $\sigma = r_S \oplus \widehat{r_1} \oplus \ldots \oplus \widehat{r_k}$ and sends σ over the out-of-band authenticated channel, and outputs g^a. For every $i \in [k]$, R_i verifies that σ is consistent with her view of the protocol. If so, R_i outputs $\widehat{D_i}^{\,b_i^{-1}}$. Otherwise, R_i outputs $\perp$ and terminates.

[16] For simplicity of presentation, we assume in this section that each pair of parties is connected via an insecure channel. The protocol can be easily adjusted to the case where each receiver shares an insecure channel only with the sender, as in Section 4.1.