Out-Of-Band Authenticated Group Key Exchange: From Strong Authentication to Immediate Key Delivery

Moni Naor
Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot 76100, Israel
http://www.wisdom.weizmann.ac.il/~naor
moni.naor@weizmann.ac.il

Lior Rotem
School of Computer Science and Engineering,
Hebrew University of Jerusalem, Jerusalem 91904, Israel
lior.rotem@cs.huji.ac.il

Gil Segev
School of Computer Science and Engineering,
Hebrew University of Jerusalem, Jerusalem 91904, Israel
segev@cs.huji.ac.il

Abstract

Given the inherent ad-hoc nature of popular communication platforms, *out-of-band authenticated key-exchange protocols* are becoming widely deployed: Key exchange protocols that enable users to detect man-in-the-middle attacks by manually authenticating one short value. In this work we put forward the notion of *immediate key delivery* for such protocols, requiring that even if some users participate in the protocol but do not complete it (e.g., due to losing data connectivity or to other common synchronicity issues), then the remaining users should still agree on a shared secret. A property of a similar flavor was introduced by Alwen, Coretti and Dodis (EUROCRYPT '19) asking for immediate decryption of messages in user-to-user messaging *while assuming that a shared secret has already been established* – but the underlying issue is crucial already during the initial key exchange and goes far beyond the context of messaging.

Equipped with our immediate key delivery property, we formalize strong notions of security for out-of-band authenticated group key exchange, and demonstrate that the existing protocols either do not satisfy our notions of security or are impractical (these include, in particular, the protocols deployed by Telegram, Signal and WhatsApp). Then, based on the existence of any passively-secure key-exchange protocol (e.g., the Diffie-Hellman protocol), we construct an out-of-band authenticated group key-exchange protocol satisfying our notions of security. Our protocol is inspired by techniques that have been developed in the context of fair string sampling in order to minimize the effect of adversarial aborts, and offers the optimal tradeoff between the length of its out-of-band value and its security.

2012 ACM Subject Classification Security and privacy → Cryptography; Theory of computation → Cryptographic protocols; Theory of computation → Cryptographic primitives

Keywords and phrases End-to-end encryption, out-of-band authentication, key exchange

Digital Object Identifier 10.4230/LIPIcs.ITC.2020.9

Related Version A full version of the paper is available at https://eprint.iacr.org/2019/1458.

Funding *Moni Naor*: Supported in part by a grant from the Israel Science Foundation (no. 950/16). Incumbent of the Judith Kleeman Professorial Chair.
Lior Rotem: Supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities and by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253).
Gil Segev: Supported by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253).

1st Conference on Information-Theoretic Cryptography (ITC 2020).
Editors: Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs; Article No. 9; pp. 9:1–9:25
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A fundamental challenge in cryptography is that of generating shared secrets in communication networks that are susceptible to man-in-the-middle attacks. When a public-key infrastructure is available, this task has been thoroughly studied, and many protocols have been suggested (see Section 1.2). The question remains, however, of how to agree on an initial secret when connections are formed ad-hoc, and a public-key infrastructure is impractical to maintain. Such scenarios include, for example, communication platforms offering end-to-end encrypted messaging services, audio calls or video calls [50, 60, 63, 62, 5, 25, 10, 20, 35, 51, 30, 19, 23, 57, 2, 31], as well as secure pairing of IoT devices (e.g., [22, 37, 11]).

Out-of-band authenticated key exchange. Given that man-in-the-middle attacks are impossible to detect without any additional setup, one approach often taken is to provide users/devices with the ability to communicate "out-of-band", assuming that they have access to an external channel through which they can information-theoretically authenticate short values. Equipped with such an external channel, one can then rely on out-of-band authenticated key-exchange protocols: Protocols that are tailored to using both a standard insecure channel and a low-bandwidth out-of-band channel, and enable users to bootstrap the limited resource of information-theoretical authentication provided by the out-of-band channel in order to establish shared secrets while detecting man-in-the-middle attacks. Such an approach is taken by most communication platforms providing end-to-end encryption and by protocols for pairing of IoT devices (see the references above).

The out-of-band channel typically corresponds to having the users compare with each other a short value displayed by their devices (or having a single user compare a string displayed by all paired devices in the context of pairing of IoT devices), but can in fact be based on a variety of real-world assumptions (e.g., [43, 27, 56, 42, 32, 22, 37, 59, 63]). In most implementations the "manual" flavor of the out-of-band channel introduces a tradeoff between the effort invested by the users and the security guarantees: A longer out-of-band value may enable better security in principal, but also incurs a more intensive user effort, thus hurting usability and ultimately security.

Non-interactive vs. interactive protocols. As in standard key exchange, there are two main flavors of out-of-band authenticated key-exchange protocols: *Non-interactive* protocols in which each user sends at most one message and this message is sent independently of the other users' messages, and *interactive* protocols in which users may send more than one message and these messages may depend on other users' previously-sent messages.[1]

Non-interactive protocols are widely used by messaging platforms (e.g., WhatsApp and Signal [63]), since they do not require any two users to be online at any particular point in time. However, such protocols are inherently limited in the security they can provide – as we discuss in Section 1.3.

[1] These two flavors of protocols are sometimes referred to as "asynchronous" protocols vs. "synchronous" protocols (e.g., in the specific context of messaging protocols [52, 19, 2, 31] – to which we do not at all limit ourselves in this work). As discussed below we follow the more standard terminology of non-interactive protocols vs. interactive protocols since the standard model of synchronous computation in distributed computing is much more restrictive than the standard model required for interactive cryptographic protocols in general, and for the protocols considered in this paper in particular (e.g., a global clock synchronizing the entire execution of the protocol among the various parties is not required [41, 3]).

Our focus: Immediate key delivery in interactive protocols. In various popular scenarios, such as voice and video calls or pairing of IoT devices, the users or devices participating in the protocol are typically expected to remain online throughout its execution. In these scenarios, unlike in messaging applications, interactive protocols may be used in order to ensure stronger security guarantees (e.g., security which is independent of the adversary's concrete running time). This is the case, for example, in the out-of-band key-exchange protocol Telegram uses for its voice calls [59].

An additional approach to constructing interactive out-of-band authenticated key exchange protocols is to first run any passively-secure key-exchange protocol, and then use an *out-of-band message authentication protocol* in order to authenticate its transcript [61, 49, 54]. An out-of-band message authentication protocol allows for the authentication of long messages while using the out-of-band channel only to information-theoretically authenticate one short value. Although any solution to the general task of establishing shared keys must inherently rely on computational assumptions, out-of-band *message authentication* protocols may provide unconditional information-theoretical security. By now there is a sound theoretical understanding (i.e., protocols and matching lower bounds) of out-of-band message authentication protocols, in both the user-to-user and the group settings, as well as practically-relevant protocols in both settings [61, 48, 46, 54, 44] – these works were indeed motivated by the task out-of-band authenticated key exchange.

In contrast, out-of-band authenticated key exchange has been studied in the user-to-user setting (e.g., [49, 40]) but has been left without any rigorous treatment in the group setting. In particular, when considering the security of out-of-band authenticated key exchange in the group setting, a crucial requirement is that even if some users participate in the protocol but do not complete it, then the remaining users should still agree on a shared key that will enable them to start interacting in an end-to-end encrypted manner. We refer to this property as *immediate key delivery*. Alwen, Coretti and Dodis [2] have recently suggested a property of such flavor to which they referred to as "immediate decryption". Their work was in the context of messaging protocols *assuming that a shared secret key has already been established* – but the underlying issue is crucial already during the initial key exchange.

Providing immediate key delivery is a challenge that arises only in the interactive setting, as it is trivially guaranteed by any non-interactive protocol (but, as discussed above, such protocols provide somewhat weaker security guarantees). Although interactive protocols are suitable for scenarios in which users are typically expected to remain online, protocols still have to address cases where some of the users do not complete the protocol. Otherwise, for example, any user who loses connectivity prevents the successful completion of the protocol by the remaining users. Moreover, if a protocol does not offer immediate key delivery, then it becomes very easy for an attacker to prevent the users from agreeing on a shared secret, by simply blocking all outgoing communication from a single user in the group.

The significant and practical importance of immediate key delivery, together with various other security considerations for out-of-band protocols, motivate an in-depth examination of out-of-band authenticated key exchange, including formal definitions and protocols that satisfy them.

1.1 Our Contributions

Motivated by the above-described state of affairs, we present the following contributions:

- We suggest a framework for analyzing out-of-band authenticated group key-exchange protocols, capturing crucial security and functionality properties that arise in the group setting for out-of-band protocols.

- We observe that the existing approaches for constructing out-of-band authenticated key-exchange protocols either do not satisfy our (standard) notions of security or are impractical (already for rather small groups). This situation highlights the fact that it is highly non-trivial to satisfy our notions of security while keeping the out-of-band value short.
- Based on the existence of any passively-secure user-to-user key-exchange protocol (e.g., the Diffie-Hellman protocol), we construct an out-of-band authenticated group key-exchange protocol satisfying our notions of security, and offering the optimal tradeoff between the length of its out-of-band value and its security. Moreover, for some possible use-cases, instantiating our protocol in the random-oracle models leads to a concrete and efficient protocol.

In what follows we briefly discuss each of these contributions, and the reader is referred to Section 1.3 for a more elaborate and technical overview.

Modeling out-of-band authenticated group key exchange. We consider a group of users communicating over a completely-insecure channel that is susceptible to man-in-the-middle attacks, and in addition assume that *some* user of the group can information-theoretically authenticate one short value to all other users who have not yet aborted, over the out-of-band channel (note that we do not make any assumptions as to the particular identity of that user).[2]

Within this communication model (which we formally define in Section 2), we put forth a realistic framework and notions of security for out-of-band authenticated key-exchange in the group setting, considering the following three requirements:

- **Pseudorandomness:** If a man-in-the-middle adversary does not interfere with the communication, the resulting shared key should be computationally indistinguishable from an independent and uniformly-distributed key given the transcript of the protocol *which includes the out-of-band value.*
- **Man-in-the-middle detection:** If a man-in-the-middle adversary does interfere with the communication, this should be detected except with probability $\epsilon(\lambda) + \mathsf{negl}(\lambda)$, where ϵ is a pre-determined function of the security parameter $\lambda \in \mathbb{N}$, and negl is a negligible function which may depend on the adversary.

  Most importantly, ϵ must be fixed for all adversaries, and in particular it is not allowed to depend on the adversary's on-line or off-line running time or space usage – as the *effective length* of the out-of-band value might not always be sufficiently long (e.g., when executed by "lazy users" who may not consider the out-of-band value in its entirety [44]).
- **Immediate key delivery:** Even if a subset of the parties aborts the execution of the protocol before its completion, the remaining parties should still agree on a shared key (the abort decisions may be determined adversarially throughout the execution of the protocol). This requirement significantly strengthens the standard correctness requirement of key-exchange protocols, and achieving this requirement is the core technical contribution of our work.

[2] The way that the out-of-band value is propagated through the group might be different; e.g., if some users recognize the voice of one user in a voice group call, and the other users recognize the voice of another user, then informing all users of the out-of-band value requires the two recognized users to read it out loud. Our model, in which there is a single out-of-band value and a single user who sends it, can always be easily translated to such situations (e.g., by having an out-of-band channel from each of the said users to the users in the group who recognize her voice).

Note that the pseudorandomness and man-in-the-middle detection requirements are relevant already in the user-to-user setting (and we consider natural extensions of these requirements from passively-secure protocols to out-of-band protocols), and that the immediate key delivery is a new requirement that we introduce in the group setting.

Existing protocols do not meet our requirements. We show that even though the three requirements listed above seem fairly standard as far as cryptographic definitions go, they are not met by existing protocols. Namely, we observe that each of the out-of-band authenticated key-exchange protocols deployed by Signal, WhatsApp and Telegram, and that the protocol suggested by Rotem and Segev [54] does not satisfy at least one of the aforementioned requirements.

Already in the user-to-user setting, we show that the protocol deployed by Telegram does not satisfy our pseudorandomness requirement, and that the protocols deployed by Signal and WhatsApp do not satisfy our man-in-the-middle detection requirement. In the group setting, even though these protocols provide immediate key delivery, they are non-scalable in terms of the length of the out-of-band value, since they require running a user-to-user protocol with each member of the group separately, resulting in an out-of-band value whose length depends linearly on the size of the group. For example, in a group of size 32, in order to get 60 bits of security, the out-of-band value in these protocols has to be of length at least $31 \times 60 = 1860$ bits (i.e., the initiator of the key exchange has to compare at least 560 decimal digits with other users). In the group setting, the protocol of Rotem and Segev, which relies on the above-mentioned transcript-authentication approach [49] is more practical, and satisfies our pseudorandomness and man-in-the-middle detection properties, but does not provide immediate key delivery.

We stress that as mentioned above, some of these protocols have their advantages in particular use cases. However, the fact that none of them provide both optimal security guarantees per our security notion and also immediate key delivery in the group setting, exemplifies in our view the difficulty that lies in satisfying all of these requirements simultaneously and highlights the challenges that need to be overcome. Looking ahead, the main reason that immediate key delivery is challenging to obtain without substantially increasing the length of the out-of-band value, is that an adversary may choose a subset of aborting users out of an *exponential* number of such subsets – and this allows the adversary significant control over the execution of the protocol.

From strong(er) message authentication to out-of-band authenticated key exchange. We construct an out-of-band authenticated group key-exchange protocol which satisfies our notions of security, based on any passively-secure user-to-user key-exchange protocol. Moreover, we prove that our protocol enjoys the optimal tradeoff (within lower-order terms) between the length of its out-of-band value and the probability of an active attack going undetected.[3]

▶ **Theorem 1** (informal). *Assuming the existence of any passively-secure user-to-user key-exchange protocol, then for any functions $n = n(\lambda)$ and $\ell = \ell(\lambda)$ there exists an out-of-band authenticated key-exchange protocol for groups of $n(\lambda)$ users, with an out-of-band value of length $\ell(\lambda)$ bits such that any active man-in-the-middle attack is detected except with probability $\epsilon(\lambda) \leq 2(n(\lambda) - 1) \cdot (1/2 + o(1))^{\ell(\lambda)}$, where $\lambda \in \mathbb{N}$ in the security parameter.*

[3] Our protocol provides such an optimal tradeoff even when executed by "lazy users", who may not consider the out-of-band value in its entirety, as recently formalized by Naor et al. [44].

Our protocol is based on a general transformation that takes any passively-secure key-exchange protocol and produces an out-of-band authenticated key-exchange protocol. Concretely, we observe that although the above-mentioned transcript-authentication approach (i.e., using a group out-of-band message authentication protocol in order to authenticate the transcript of a passively-secure group key-exchange protocol) fails to guarantee immediate key delivery, this can be overcome if the underlying message authentication protocol provides a property we refer to as *immediate message delivery* (the precise transformation requires overcoming various additional challenges). We construct such a strengthened out-of-band message authentication protocol by starting from the basic structure of the group protocol of Rotem and Segev, and incorporating within it techniques from the realm of fair multi-party string-sampling protocols (i.e., protocols in which even if some parties abort then the remaining parties sample a "relatively unbiased" string [4, 18] – see Section 1.3 for more details). We view this as our main technical contribution.

A benefit of the fact that we present our protocol as a general transformation while relying on generic building blocks, is that this enables for a much greater modularity in its instantiation. In particular, this allows for the reliance on post-quantum secure assumptions as opposed to the currently deployed protocols by Telegram, Signal and WhatsApp that are based on the Decisional Diffie-Hellman assumption.

1.2 Related Work

The problem of detecting man-in-the-middle attacks in key exchange protocols has been studied extensively in various models (see, for example, [8, 6, 58, 7, 16, 36] for user-to-user protocols, and [9, 15, 34] for group protocols). Our setting and definitions bear some resemblance in particular to that of password-authenticated key exchange (PAKE; see [29, 12, 33, 26, 1] and the references therein), in that in both cases the security is inherently a function of the unpredictability of some short value (the out-of-band value in our case, and the shared password in the case of PAKE).

In particular, in the PAKE setting, Fiore, Vasco and Soriente [24] considered the problem of "partitioned group key exchange" which is conceptually somewhat similar to the problem we consider in this paper: Designing a PAKE protocol with the guarantee that even if some users provide a wrong password then all users who provided the correct password should still agree on a shared key. The main difference, however, between this problem and our work is the correctness requirement: Fiore et al. assume that all users are on-line and follow the instructions of the protocol, and require that all users who provide the same password output the same key, whereas we assume that some users may adversarially abort the protocol at any stage and require that all other users output the same key. This difference, together with the substantial differences of the two authentication models, lead to completely different technical challenges (and solutions).

More generally, although there are natural similarities between the various authentication models, there are several key differences between our work and the lines of works mentioned above. Namely, to provide immediate key delivery, our model and definitions accommodate users who abort prematurely, whereas most of the works on authenticated key-exchange are either in the user-to-user setting, or consider groups that remain static (i.e., no users are added or removed) *throughout the execution of the protocol.* Some works (e.g., [13, 14]) do consider dynamic groups that may change over time and their shared secret needs to be updated, but not the scenario that we are studying of users who abort *during the execution of the protocol itself.* In that respect, our work is focused on initial key exchange (and its authentication), and we do not explicitly consider the task of adding or removing users in later stages. In any case, adding a user to the group while communicating with only a single existing member of the group, as is the case with the deployed protocols, can and must be authenticated using a user-to-user out-of-band protocol. This approach can also be used to add users who went offline during the initial setup, which again must require an additional out-of-band verification.

In the out-of-band model, most previous works concentrated on message authentication [53, 61, 46, 48, 54, 44], with the exception of Pasini and Vaudenay [49] and Lindell [40], who studied key exchange explicitly, but only in the user-to-user setting. Pasini and Vaudenay followed the transcript-authentication paradigm described above, while Lindell focused on analyzing the specific Bluetooth v2.1 comparison-based key-exchange protocol.

1.3 Overview of Our Security Notions and Construction

In this section we first discuss the motivation underlying our three security requirements (which were briefly mentioned in Section 1.1 and are formally defined in Section 3). Next, we overview the "transcript authentication" approach for constructing an out-of-band authenticated group key-exchange protocol (which serves as our starting point), and point out its current limitations. Then, we provide a high-level overview of our construction and of its proof of security.

Our notions of security. Our work puts forward extensions of the standard notions of pseudorandomness and man-in-the-middle detection that are tailored to out-of-band protocols, as well as introduces the notion of immediate key delivery, as discussed in Section 1.1.

Requirement 1: Pseudorandomness given the out-of-band value. The out-of-band channel is assumed to provide authenticity for one short value, but it is not assumed to provide any form of secrecy, and thus all communication over this channel may be completely visible to an adversary. Thus, the natural extension of the standard pseudorandomness requirement for key-exchange protocol must consider an adversary observing both the communication over the insecure channel and over the out-of-band channel. For such an adversary, the resulting shared key should be computationally indistinguishable from an independent and uniformly-distributed key.

Requirement 2: Adversary-independent man-in-the-middle detection. The probability of detecting an active man-in-the-middle attack depends (at least) on the bit-length ℓ of the out-of-band authenticated value (in Section 3 we provide a simple proof showing that any protocol can be undetectably attacked with probability essentially $\epsilon = n \cdot 2^{-\ell}$). We require that active attacks are detected with probability that depends on the protocol itself (e.g., $\epsilon = n \cdot 2^{-\ell}$), and do not scale in a meaningful manner with the adversary's on-line or off-line running time or space usage. For example, our requirement rules out protocols that out-of-band authenticate an 80-bit value, and an adversary that can execute 2^{40} computations of a certain hash function can break its security with probability $2^{40} \cdot 2^{-80}$. This property is even more crucial when considering the likely scenario of "lazy users", as formalized by Naor et al. [44], where users may consider only a short sub-string of the out-of-band authenticated value. This renders the "effective length" of the out-of-band value much shorter than its actual length ℓ. For example, if the security that a protocol provides is $T \cdot 2^{-\ell}$, where T is roughly the running time of the adversary and ℓ is the length of the "de-facto out-of-band value", then if the users consider, say 20 bits from the out-of-band value, an adversary running in reasonable time can break the security of the protocol quite easily (instead of having the protocol still guarantee the best-possible security of $\epsilon = 2^{-20}$).

Requirement 3: Immediate key delivery. We require that even if a subset of the parties aborts the execution of the protocol before its completion, the remaining parties should still agree on a shared key. This is a crucial requirement not only due to the above-described nature of mobile-based messaging, but even more in order to protect against devastating adversarial denial-of-service attacks that are undetected by other users. For example, in the recently-suggested protocol of Rotem and Segev [54], an adversary that can simply block the communication going out of just one user, can make sure that the other users will never agree on a shared key, leaving the group either completely vulnerable or utterly useless.

Although this property is a functionality-focused one, our main technical challenge in this work is to obtain it while retaining a good (and preferably optimal) level of security. As we discuss in length in the continuation of this section, simple attempts to add immediate key delivery to the protocol of Rotem and Segev make it completely insecure.

Interaction is essential. Satisfying all three requirements simultaneously requires an interactive protocol. The pseudorandomness requirement may be satisfied both by interactive and by non-interactive protocols (under suitable assumptions). The third requirement, immediate key delivery, is trivially satisfied by any non-interactive protocol, but as mentioned above, the second requirement – adversary-independent MitM detection – cannot be satisfied by such protocols. Concretely, in Section 3 we show that for any non-interactive protocol and for any running time T, there exists a successful man-in-the-middle attacker that runs in time essentially T and is undetected with probability $\min\{1/3, \Omega(T \cdot 2^{-\ell})\}$, where ℓ is the bit-length of the out-of-band value. In this light our goal is to come up with interactive protocols that simultaneously guarantee all three requirements, while retaining a short out-of-band value.

Our starting point: The "transcript authentication" approach. As mentioned in Section 1.1, the out-of-band group key-exchange protocols deployed by WhatsApp, Signal and Telegram provide immediate key delivery, but impose a heavy burden on the users: These protocols require running a user-to-user protocol with each member of the group separately, resulting in an out-of-band value whose length depends linearly on the size of the group. In addition, recall that these protocols do not satisfy our two additional security requirements, and thus they do not seem to be promising starting points for designing protocols satisfying our goals.

Our starting point is the transcript-authentication approach described above [49], while using the out-of-band group message authentication protocol of Rotem and Segev [54]. Roughly speaking, this approach suggests running any passively-secure group key-exchange protocol,[4] and afterwards to authenticate its transcript via the following out-of-band message authentication protocol:

1. P_1 chooses $r_S \leftarrow \{0,1\}^\ell$ and commits to $\mathsf{trans} \| r_S$ to all other users, where trans is the transcript of the key-exchange protocol from P_1's point of view.
2. $P_2, \ldots, P_n$ cooperatively choose a string r_R: Each P_i chooses $r_i \leftarrow \{0,1\}^\ell$ and commits to it to all other users. After all users have committed, each P_i decommits to reveal r_i, and sets $r_R = \bigoplus_{i \in \{2,\ldots,n\}} r_i$.

[4] Most naively, the initiator P_1 can execute a user-to-user protocol (such as the Diffie-Hellman protocol) with each other user P_i for obtaining a shared key k_i. Then, P_1 will sample a random key k and encrypt it to each other user P_i using the key k_i.

3. P_1 decommits to reveal r_S, and then out-of-band authenticates to $\sigma = r_S \oplus r_R$. Each of the other users accepts (and outputs the key agreed upon in the key exchange step) if and only if σ and trans are both consistent with her view.

This protocol falls short of satisfying our definition for out-of-band authenticated group key exchange in two respects. First, our definition requires that an active attack will be detected except with some pre-determined probability, but the only guarantee provided by the protocol of Rotem and Segev is that if trans is inconsistent with the view of some P_i, then with high probability this P_i will reject. It might still be the case though, that an active adversary modifies messages sent during the out-of-band message authentication phase described above.

This problem may be addressed in a simple manner (and in this specific protocol it is not that devastating to begin with): Instead of using the out-of-band message authentication protocol in order to authenticate the transcript trans of the group key exchange, P_1 samples a pair $(\mathsf{sk}, \mathsf{vk})$ of signing and verification keys for a *one-time strongly unforgeable signature scheme*; then uses the out-of-band message authentication protocol to authenticate vk to the other users; and finally uses sk to sign the transcripts of *both* the key-exchange protocol and the out-of-band message authentication protocol.

The second, more fundamental, problem is that the protocol of Rotem and Segev does not provide "immediate key delivery", even if the underlying passively-secure key-exchange protocol does provide it[5]. This is true since a user who identifies a deviation from the protocol (including a premature abort) terminates and rejects. In order for the out-of-band authenticated group key-exchange protocol to provide immediate key delivery, the out-of-band message authentication protocol needs to satisfy a similar property, to which we refer as *immediate message delivery*. This property essentially requires that even if a subset of the receivers in the protocol abort, but the execution is otherwise honest, the rest of the receivers should still accept the message.

Alas, the lacuna in the out-of-band message authentication protocol of Rotem and Segev, due to which it does not provide immediate message delivery, is far from being a mere technicality. To see why, consider what happens if we simply ignore aborting users, and take r_R to be the exclusive-or of only the r_i's of the users who opened their commitments. This might provide immediate key delivery, but gravely hurts the security of the protocol, by giving the man-in-the-middle adversary the ability to choose which commitments to open to each P_i *after observing* r_i. Concretely, in the full version of this paper [45], we present an attack showing that this change exponentially increases the forgery probability from roughly $n \cdot 2^{-\ell}$ to roughly $2^n \cdot 2^{-\ell}$, where n is the number of users in the group and ℓ is the length of the out-of-band value.

The underlying issue with the protocol of Rotem and Segev (explaining the exponential increase), is that a man-in-the-middle adversary interacting with, say P_i, can choose to abort any subset of $\{P_2, \ldots, P_n\} \setminus \{P_i\}$ towards P_i, before forwarding the decommitments of the users in this subset to P_i. Even if the interaction with P_i is otherwise honest, each possible aborting subset might induce a different value for r_R in the view of P_i. This enables a man-in-the-middle adversary to substantially "steer" the r_R that P_i computes, such that the attack will go undetected.

[5] The naive protocol described in Footnote 4 is a passively-secure protocol with immediate key delivery: Even if some user aborts then the remaining users still output the key k chosen by P_1.

Providing immediate message delivery: Attempt I. As a first attempt to limit the additional power provided to the adversary by allowing aborts, consider a "restart-after-abort" variant of the Rotem-Segev protocol, in which after an abort by any of the users, the remaining users start a fresh execution of the protocol. Intuitively, now the adversary has no incentive to abort more then a single user in each execution of the original Rotem-Segev protocol, and the identity of the particular user who aborts (if such a user exists) is of no consequence due to the symmetry of the protocol. Hence, instead of exponentially many choices of aborting subsets, in each execution of the original Rotem-Segev protocol the adversary effectively has only two (abort or not).

The problem with this approach however, is that now the adversary has up to $n - 1$ attempts to break the security of the Rotem-Segev protocol, yielding a forgery probability of roughly $n^2 \cdot 2^{-\ell}$. This is much better than the $2^n \cdot 2^{-\ell}$ forgery probability of the "vanilla" Rotem-Segev protocol, but still quite far from optimal: The forgery probability grows quadratically with the number of users in the group, which may be significant in large groups, and as we show below, this can be avoided. Moreover, when the protocol is executed by lazy users as discussed above (who may not consider the out-of-band value in its entirety [44]), the effective value of ℓ might be relatively small, resulting in a substantial forgery probability. Instead, we are interested in a solution that provides security which is optimal with respect to the size of the group and to the length of the out-of-band value, so that it provides reasonable security even for lazy users (looking ahead, our protocol provides the *optimal* tradeoff within lower-order terms between the length of its out-of-band value and its security even when executed by lazy users).

Providing immediate message delivery: Attempt II. In light of the above, and inspired by techniques from protocols for fair string sampling, we construct a group out-of-band message authentication protocol that provides immediate message delivery – while retaining an optimal level of security (within lower order terms). The main idea behind our protocol is to replace the manner r_R is chosen in the protocol of Rotem and Segev, with a way which is more resilient to aborts. By that, intuitively speaking, we mean that even a man-in-the-middle adversary interacting with some P_i, and can simulate control over all users but P_i in that interaction, cannot force the r_R computed by P_i to hit the particular value that it needs in order for the attack to go unnoticed by P_i.

Instead of selecting r_R in "one shot" as done in the protocol of Rotem and Segev, in our protocol it is chosen in more gradual manner, which considerably limits the effect of adversarial aborts. Concretely, the users iteratively choose T ℓ-bit values $r_{R,1}, \ldots, r_{R,T}$ (where T is a parameter of the protocol) one after the other, in T consecutive iterations. In the tth iteration $r_{R,t}$ is chosen by the remaining users among $P_2, \ldots, P_n$ (i.e., the users who have not yet aborted) in the same manner as r_R is chosen in the protocol of Rotem and Segev. Finally, the value of r_R in our protocol is then taken to be the bit-wise majority of $r_{R,1}, \ldots, r_{R,T}$: The kth bit of r_R is the majority bit over the kth bits of $r_{R,1}, \ldots, r_{R,T}$. We refer the reader to Sections 4 and 5 for a complete and formal description of our protocol.

With this change, analyzing our new protocol proves to be technically involved, as a man-in-the-middle adversary has numerous more possible "synchronizations" (i.e., different orderings of messages) to impose on an execution of the protocol. Nevertheless, we manage to prove that when the commitment scheme used in our protocol is statistically-binding and concurrent non-malleable,[6] then the forgery probability is bounded roughly by $n \cdot (1/2 + n/\sqrt{T})^{\ell}$. Setting

[6] see the full version [45] as well as [21, 39, 28, 17] and the references therein for further details on such

the parameter T to be $n^2 \cdot \omega(1)$ (e.g., $n^2 \cdot \log^* \lambda$), we get that the forgery probability is $n \cdot (1/2 + o(1))^\ell$, matching our lower bound of $\min\{1/3, \Omega(n \cdot 2^{-\ell})\}$ (see Section 3) within lower order terms.

Overview of our proof of security. We provide a brief and high level overview of the proof of unforgeability of our out-of-band message authentication protocol, ignoring various technical difficulties and focusing on the main ideas. We prove that for every $i \in \{2, \ldots, n\}$, if the man-in-the-middle changes the verification key sent to P_i in the beginning of our out-of-band authenticated key-exchange protocol,[7] then the probability that P_i will not detect this interference (i.e., will not output $\perp$) is upper bounded by roughly $(1/2 + n/\sqrt{T})^\ell$. We do so by considering all possible synchronizations that a man-in-the-middle might impose on an execution of the protocol relative to P_i, and bound the probability of forgery in each of them relying on the statistical binding and on the concurrent non-malleability of the underlying commitment scheme. We manage to partition all possible such synchronizations into two families, and handle each one separately. For simplicity of presentation in this overview, we focus on the case where $\ell = 1$ (i.e., the initiator P_1 out-of-band authenticates a single bit), and the reader is referred to Section 5 for our formal proof of security.

Proof of security: Case I. In the first family of synchronizations, P_1 decommits to reveal r_S before P_i receives the first round of commitments from $P_2, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$. In this case, by the statistical binding, the values of r_S and r_R according to the view of P_1 and the value of r_S according to the view of P_i, have all been determined by the time P_i receives the first round of commitments. Hence, in order for P_i to not reject, the man-in-the-middle must make sure that r_R according to the view of P_i hits the unique value $r_R^* \in \{0, 1\}$ which is the exclusive-or of the three aforesaid determined values. We bound the probability that $r_R = r_R^*$ using the concurrent non-malleability of the commitment scheme, where the heart of the proof lies in two parts.

The computational part, in which we show that no strategy of the man-in-the-middle can result in a noticeably-greater probability that $r_R = r_R^*$ then the following strategy, denoted M_{opt}: (1) In each iteration $t \in [T]$ and for every P_j that has not yet aborted according to the view of P_i, send P_i a commitment to the value 1 from P_j; (2) If the sampled value in this round $r_{R,t}$ is equal to r_R^* (when no user aborts), then open all commitments; (3) Otherwise, open all commitments except for that of the minimal-index user P_j that has not yet aborted (since P_j committed to the value 1, this is guaranteed to flip the bit $r_{R,t}$, so that it is equal to r_R^*).

We prove that this strategy is optimal in forcing $r_R = r_R^*$ (within a negligible additive factor) via a hybrid argument: We start with any other man-in-the-middle adversary M and gradually change its strategy to M_{opt}, iteration by iteration, proving that the probability that $r_R = r_R^*$ cannot decrease by too much in each change, or the concurrent non-malleability of the commitment scheme is violated. Concretely, we consider $T + 1$ hybrids, where the adversary in the tth hybrid, denoted by M_t, plays as M in the first $T - t$ iterations and as M_{opt} in the remaining t iterations. Observe, that in order for M_t to succeed with noticeably-greater probability than M_{t+1} (in forcing $r_R = r_R^*$), it must be the case that in the tth hybrid,

commitment schemes.

[7] Our protocol in Section 5 is a general-purpose out-of-band message authentication protocol. For concreteness in this overview, we focus on the case where the message to be authenticated is the verification key sampled by P_1, as is the case in our out-of-band authenticated key-exchange protocol.

$\Pr[r_{R,t} = r_R^*]$ is noticeably greater than 1/2. This contradicts the concurrent non-malleability of the underlying commitment scheme: Intuitively, this is because in an ideal experiment in which the bit contributed by P_i in the tth iteration is sampled anew just before P_i decommits, it holds that $\Pr[r_{R,t} = r_R^*] = 1/2$.

The statistical part, in which we show that the optimal adversary described above, M_{opt}, succeeds in forcing $r_R = r_R^*$ with probability no greater than roughly $1/2 + n/\sqrt{T}$. To prove this, it is convenient to think of an equivalent experiment, in which $r_{R,1}, \ldots, r_{R,T}$ are first sampled uniformly from $\{0,1\}$, and then the adversary is given the option to flip $n-2$ of them.[8] In this experiment, the adversary can force $r_R = r_R^*$ if and only if $|\{t \in [T] : r_{R,t} = r_R^*\}| \geq T/2 - n + 2$. We observe that $|\{t \in [T] : r_{R,t} = r_R^*\}|$ is a random variable distributed according to the binomial distribution with parameters 1/2 and T. We then use the symmetry of this distribution and the fact that every value in its support is obtained with probability no greater than roughly $1/\sqrt{T}$, in order to bound the probability that $|\{t \in [T] : r_{R,t} = r_R^*\}| \geq T/2 - n + 2$ by roughly $1/2 + n/\sqrt{T}$.

Proof of security: Case II. In the second family of possible synchronizations, P_1 decommits to reveal r_S after P_i has received at least one round (and possibly many) of commitments from the other users. Denote the last round of commitments received by P_i before P_1 decommits by $t^* \in [T]$. In this case the man-in-the-middle adversary's situation is worse than in Case 1: The hiding and the concurrent non-malleability of the commitment scheme, and in particular of the commitment by P_1 to the value r_S, imply that the adversary cannot hope to force any of $r_{R,1}, \ldots, r_{R,t^*}$ to be equal to r_R^* with probability noticeably greater than 1/2. This is because in an ideal experiment in which r_S is sampled anew just before P_1 decommits, it holds that $\Pr[r_{R,t} = r_R^*] = 1/2$ for every $t \in [t^*]$ and independently of the other rounds, and irrespective of the identity of the aborted users in rounds $1, \ldots, t^*$. Intuitively speaking, it follows that the adversary is only more limited than in the previous case, as she can use her "abort quota" effectively only in rounds $t^* + 1, \ldots, T$, and hence the forgery probability in this case cannot be noticeably greater than that of the previous case.

1.4 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we review the out-of-band communication model, and in Section 3 we present our notions of security for out-of-band authenticated group key-exchange protocols. In Section 4 we show that any passively-secure user-to-user key-exchange protocol can be transformed into an out-of-band authenticated group key-exchange protocol that satisfies our notions of security, and in Section 5 we construct an out-of-band message authentication protocol with immediate message delivery, which is the main building block underlying our transformation.

2 The Out-of-Band Communication Model

In this section we review the out-of-band communication model as well as the notion of an out-of-band message authentication protocol [61, 48, 54].

[8] For the case $\ell = 1$, this is indeed equivalent. For the general case of $\ell \geq 1$, this may only add power to the adversary, and hence the probability that $r_R = r_R^*$ can only increase. Hence, in the general case as well, bounding the probability that $r_R = r_R^*$ in this experiment bounds the probability that the man-in-the-middle adversary can force $r_R = r_R^*$.

The out-of-band channel and man-in-the-middle attacks. As formalized by Vaudenay and by Naor et al. in the user-to-user setting [61, 47] and extended by Rotem and Segev to the group setting [54], interaction among users in the out-of-band communication model occurs over two types of channels: Insecure channels and a low-bandwidth authenticated channel (referred to as the "out-of-band channel"). It is assumed that a man-in-the-middle adversary has complete control over the insecure channels: The adversary can read, delay and remove messages sent by the parties over the insecure channels, as well as insert new messages at any point in time. One may consider various topologies for the network of insecure channels. For our protocols we assume the minimal such topology: An insecure channel between some user (e.g., the initiator of the protocol) and any other user in the group (i.e., a star network).

As for the out-of-band channel, it is assumed that there exists some user that can out-of-band authenticate one short value to all other users in the group. This value is assumed to be authenticated but not secret: The adversary may read or remove this message for some or all users, and may delay it for different periods of time for different users, but cannot modify it in an undetectable manner. We stress that our requirement of the out-of-band channel is a rather weak one: We only require that there exists *some* user that can out-of-band authenticate a short value to the rest of the group, and we do not apply any restrictions as to who that user is.

In addition, we do not make any synchronization assumption regarding the out-of-band channel: We do not assume that all users have to be on-line when the out-of-band value is transmitted. Specifically, any subset of the users may be off-line at that time, and any user that comes back on-line will be able to make her own decision regarding the authenticity of the execution if and when the out-of-band value reaches her (recall that the attacker can block the out-of-band value to all or to some of the users). See [55] for a more in-depth discussion of the group out-of-band communication model.

Out-of-band message authentication. An out-of-band message authentication protocol enables a sender S to authenticate a message m, which may be chosen by the adversary, to all other users $R_1, \ldots, R_n$ in the group ($n = 1$ is the user-to-user setting, whereas $n \geq 2$ is the group setting). Once the execution is completed, each receiver R_i outputs either some message $\widehat{m}_i$ or the unique symbol $\perp$ implying rejection. The following definition was introduced by Rotem and Segev [54], naturally extending those of Vaudenay and Naor et al. [61, 47]:

▶ **Definition 2.** *Let $\ell = \ell(\lambda), \epsilon = \epsilon(\lambda)$ and $n = n(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. A group (ℓ, ϵ)-out-of-band message authentication protocol for $n(\lambda)$ receivers and message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ is an $(n(\lambda)+1)$-party protocol, in which S sends at most $\ell(\lambda)$ bits over the out-of-band channel, and the following requirements hold:*

1. ***Correctness:*** *For every $\lambda \in \mathbb{N}$, for every $m \in \mathcal{M}_\lambda$ and every $i \in [n(\lambda)]$ it holds that $\Pr[\widehat{m}_i = m] = 1$, where the probability is over the randomness of the parties in an honest execution of the protocol.*
2. ***Unforgeability:*** *For every probabilistic polynomial-time adversary M there exists a negligible function $\nu(\cdot)$ such that for every input message $m \in \mathcal{M}_\lambda$ chosen by the adversary for the sender S it holds that*

$$\Pr\left[\exists i \in [n(\lambda)] : \widehat{m}_i \notin \{m, \perp\}\right] \leq \epsilon(\lambda) + \nu(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the probability is taken over the randomness of the parties and the randomness of M in an execution of the protocol with M as the man-in-the-middle adversary.

Existing out-of-band message authentication protocols. In the user-to-user setting, Vaudenay [61] constructed a protocol in which the forgery probability ϵ is upper bounded by $2^{-\ell}$, where ℓ is the bit-length of the out-of-band authenticated value, and Vaudenay and Pasini [48] proved a matching lower bound. In the group setting, considering a strengthened version of Definition 2, Rotem and Segev [54] constructed a protocol for groups of size n in which the forgery probability is bounded by $(n-1) \cdot 2^{-\ell}$, and proved a matching lower bound. Both protocols can be based on the existence of any one-way function [38, 54] via non-malleable commitments.

3 Out-of-Band Authenticated Group Key Exchange

In this section we first present our strengthened notion of security for out-of-band authenticated key-exchange protocols.[9]

Pseudorandomness, MitM Detection and Immediate Key Delivery. Our strengthened notion of security for out-of-band key-exchange protocols consists of three requirements: Pseudorandomness and man-in-the-middle detection that are relevant already in the user-to-user setting, and immediate key delivery that we introduce in the group setting (as discussed in Section 1.3). Our pseudorandomness and man-in-the-middle detection requirements are natural extensions of these requirements to the out-of-band model:

- If a man-in-the-middle adversary does not interfere with the communication, the resulting shared key should be computationally indistinguishable from an independent and uniformly-distributed key given the transcript of the protocol *which includes the out-of-band value.*
- If a man-in-the-middle adversary does interfere with the communication, this should be detected except with probability $\epsilon(\lambda) + \mathsf{negl}(\lambda)$, where ϵ is a pre-determined function of the security parameter $\lambda \in \mathbb{N}$, and negl is a negligible function which may depend on the adversary. Most importantly, ϵ must be fixed for all adversaries (e.g., it is not allowed to depend on the adversary's on-line or off-line running time or space usage).

  Our security definition requires that an active attack is detected by all users on the receiving end of the out-of-band channel, for whom communication to or from them has been actively modified by the attacker. The task of notifying all other users (who are still online at the end of the execution) of an active attack can be achieved, for example, by assuming that all users can send an "out-of-band feedback" signal to all other members, indicating an attack. Observe that such an assumption (or an assumption of the same nature) is essential in order for all users to detect an active attack, as without it (i.e., with only a single user that can send a message out-of-band and all other communication being subject to man-in-the-middle manipulation) an active attack in which some of the users do not identify the attack is always possible.

Our immediate key delivery requirement significantly strengthens the standard correctness requirement of key-exchange protocols: Even if a subset of the parties aborts the execution of the protocol before its completion, the remaining parties should still agree on a shared

[9] In the full version [45], we show that the protocols deployed by Signal, WhatsApp and Telegram do not satisfy it already in the user-to-user setting, and that the protocol obtained via the "out-of-band transcript authentication" approach does not satisfy it in the group setting. We also show that there is a simple and practically-relevant user-to-user protocol that does satisfy our notion of security (and offers the optimal trade-off between the length of its out-of-band authenticated value and its man-in-the-middle detection probability).

key. To capture this requirement, for an algorithm A and an n-party protocol π, we let $\mathsf{FailStopExec}(\pi, A, \lambda)$ denote the output of the following experiment:

1. Start an execution of π with joint input 1^λ.
2. For every $i \in [n] \setminus \{1\}$, before P_i sends a message v according to π, invoke $\mathsf{decision} \leftarrow A(1^\lambda, \mathsf{PartialTrans})$, where $\mathsf{decision} \in \{\mathsf{abort}, \mathsf{continue}\}$ and $\mathsf{PartialTrans}$ is the partial transcript of the execution up to this point. If $\mathsf{decision} = \mathsf{continue}$, P_i sends v, and the execution continues. If $\mathsf{decision} = \mathsf{abort}$, P_i aborts and the execution continues without P_i.
3. The output of $\mathsf{FailStopExec}(\pi, A, \lambda)$ is a $(n+1)$-tuple $(\mathsf{AbortSet}, \mathsf{v}_1, \dots, \mathsf{v}_n)$, where $\mathsf{AbortSet}$ denotes the set of indices of aborted parties at the end of the execution and v_i is the output of P_i if $P_i \notin \mathsf{AbortSet}$ and $\mathsf{v}_i = \perp$ otherwise.

Note that in order for this experiment to be well defined, the protocol π has to be well defined for any possible pattern of aborts. In that case, this experiment is well defined both for group key-exchange protocols (including passively-secure ones) and for group authentication protocols (looking ahead, this experiment will enable us to formalize a notion of "immediate message delivery" for authentication protocols). When π is a key-exchange protocol, we use $\mathsf{k}_1, \dots, \mathsf{k}_n$ instead of $\mathsf{v}_1, \dots, \mathsf{v}_n$ to denote the output keys of the users.

Also note that we assume that P_1 does not prematurely abort throughout the execution. This is essential, as we will assume without loss of generality that P_1 is the user that can send a short message over the out-of-band channel. Hence, if P_1 aborts prior to sending the out-of-band value, no meaningful security can be guaranteed. Practically speaking, in the context of messaging platforms, P_1 who initiates the key-exchange protocol is typically the first party to send an encrypted message to the group. Hence, if P_1 aborts, the need for a shared key is postponed until another message is sent (at which point, the users will execute the out-of-band group key-exchange protocol when initiated by a potentially different user).

Our definition, provided below, relies on the following notation. We denote by $\mathsf{MitMExec}(\pi, M, \lambda)$ the distribution over $(n+1)$-tuples $(\mathsf{view}_M, \mathsf{k}_1, \dots, \mathsf{k}_n)$ induced by an execution of the protocol with a man-in-the-middle M, where the adversary and all parties run on input 1^λ, and view_M is the view of M at the end of the protocol ($\mathsf{k}_1, \dots, \mathsf{k}_n$ are defined as before). For every $i \in \{2, \dots, n\}$, let Active_i be the event in which the adversary actively changes the communication from or to P_i; i.e., by either modifying or removing messages sent from or to P_i or by inserting new message to or from P_i.

We also define the event Active: Informally, Active is the event in which the man-in-the-middle adversary M changes the communication among the parties in any manner that goes beyond simulating an abort by a subset of the parties (by simulating an abort by a party, we mean blocking all messages sent by that party from some point onward). More formally, let $q = q(\lambda)$ be a bound on the number of rounds in an execution of π on joint input 1^λ. For an execution according to $\mathsf{MitMExec}(\pi, M, \lambda)$, we denote by $\mathsf{Msgs}_i = (m_{i,1}, \dots, m_{i,q})$ the vector of messages sent (in order) by P_i, where if P_i has sent t messages for $t < q$, we denote $m_j = \perp$ for every $j \in \{t+1, \dots, q\}$. Similarly, we denote by $\widehat{\mathsf{Msgs}_i} = (\widehat{m_{i,1}}, \dots, \widehat{m_{i,q}})$ the vector of messages received (in order) by parties other than P_i, as messages from P_i. We denote by Active the event in which for some $i \in [n]$, there exits $t \in [p]$ such that $m_{i,t} \neq \widehat{m_{i,t}}$ and at least one of the following conditions hold: (1) $\widehat{m_{i,t}} \neq \perp$; or (2) There exists $t' > t$ such that $m_{i,t'} \neq \perp$.

▶ **Definition 3.** *Let $n = n(\lambda), \ell = \ell(\lambda)$ and $\epsilon = \epsilon(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. A group out-of-band (ℓ, ϵ)-key-exchange protocol over key space $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda \in \mathbb{N}}$ for a group of size $n = n(\lambda)$ is an n-party protocol $\pi = \langle P_1, \dots, P_n \rangle$, in which P_1 sends at most $\ell(\lambda)$ bits over the out-of-band channel and the following requirements hold:*

- ***Immediate key delivery:*** *For every $\lambda \in \mathbb{N}$ and every probabilistic polynomial-time algorithm A, it holds that*

  $$\Pr\left[\forall i \in [n(\lambda)] \setminus \mathsf{AbortSet} : \mathsf{k}_1 = \mathsf{k}_i \in \mathcal{K}_\lambda\right] = 1$$

  where $(\mathsf{AbortSet}, \mathsf{k}_1, \ldots, \mathsf{k}_n) \leftarrow \mathsf{FailStopExec}(\pi, A, \lambda)$.
- ***Man-in-the-middle detection:*** *For any probabilistic polynomial-time algorithm M there exists a negligible function $\nu(\cdot)$ such that*

  $$\Pr\left[\exists i \in \{2, \ldots, n(\lambda)\} : \mathsf{Active}_i \wedge \mathsf{k}_i \neq \bot\right] \leq \epsilon(\lambda) + \nu(\lambda)$$

  for all sufficiently large $\lambda \in \mathbb{N}$, where $(\mathsf{view}_M, \mathsf{k}_1, \ldots, \mathsf{k}_n) \leftarrow \mathsf{MitMExec}(\pi, M, \lambda)$.
- ***Pseudorandomness:*** *For any probabilistic polynomial-time algorithms M and D there exists a negligible function $\nu(\cdot)$ such that*

  $$\left|\Pr\left[\overline{\mathsf{Active}} \wedge D(1^\lambda, \mathsf{view}_M, \mathsf{k}_1) = 1\right] - \Pr\left[\overline{\mathsf{Active}} \wedge D(1^\lambda, \mathsf{view}_M, \mathsf{k}) = 1\right]\right| \leq \nu(\lambda)$$

  for all sufficiently large $\lambda \in \mathbb{N}$, where $(\mathsf{view}_M, \mathsf{k}_1, \ldots, \mathsf{k}_n) \leftarrow \mathsf{MitMExec}(\pi, M, \lambda)$ and $\mathsf{k} \leftarrow \mathcal{K}_\lambda$.

We note that when $n = 2$, Definition 3 captures the user-to-user setting. In this case, the immediate key delivery property simply reverts back to the standard correctness property of key-exchange protocols. In addition, note that the immediate key delivery property is defined with respect to an efficient algorithm A, but our construction provides immediate key delivery even in the case where A is unbounded and receives access to the random coins of the users.

Interaction is essential. As mentioned in Section 1.3, no non-interactive protocol can satisfy our man-in-the-middle detection requirement. To see why that is, let π be such a non-interactive protocol and let P_i be any user participating in the protocol (other than the one in charge of sending the out-of-band value). Consider the following man-in-the-middle attacker, that can compute the secret key outputted by P_i:

1. The attacker forwards all messages sent by the users to all users participating in the protocol, other than to P_i. Let σ be the true out-of-band value sent as a result.
2. Let m_i be the message sent by P_i. The attacker samples T independent tuples of messages $M_{-i}^{(1)}, \ldots, M_{-i}^{(T)}$ for the other users participating in the protocol, and computes the T resulting out-of-band values $\sigma^{(1)}, \ldots, \sigma^{(T)}$ (i.e., $\sigma^{(j)}$ is the out-of-band value in the execution in which the messages sent are m_i and the messages in $M_{-i}^{(j)}$).
3. If for any $j^* \in [T]$ it holds that $\sigma^{(j^*)} = \sigma$, then the attacker sends the messages in the tuple $M_{-i}^{(j^*)}$ to P_i (as the messages sent by the other users in the protocol). Otherwise, the attacker has failed and she terminates the attack.

Observe that if the attacker completes the attack, then: (1) She knows the randomness used to sample the messages in $M_{-i}^{(j^*)}$, so she can compute the key outputted by P_i; and (2) The view of P_i is the same as in an honest execution in which the messages are m_i and the messages in $M_{-i}^{(j^*)}$, so the attack is undetected by P_i.

Hence, in order to analyze the probability that this attack is successful, we need to look at the probability that there exists such an index j^*. It turns out that we can bound this probability for any choice of m_i, so let us fix m_i and look at σ and $\sigma^{(1)}, \ldots, \sigma^{(T)}$ when P_i sends m_i. These are $T + 1$ independent samples from the distribution over the out-of-band value in a random execution of π, conditioned on P_i sending the message m_i. One can verify that the probability that there exists an index $j^* \in [T]$ such that $\sigma^{(j^*)} = \sigma$ is minimized when this conditional distribution is the uniform distribution over $\{0,1\}^{\ell}$. For this distribution, the probability that there exists such an index j^* – and that the attack is successful – is at least $\min\{1/3, \Omega(T \cdot 2^{-\ell})\}$. The complete analysis is in the full version of the paper [45].

The required length of the out-of-band value. Theorem 4 states that any out-of-band group key-exchange protocol for n users with an out-of-band value of length ℓ bits can be undetectably attacked by an efficient man-in-the-middle adversary with probability roughly $n \cdot 2^{-\ell}$. As discussed in Section 1.3, a key goal in the out-of-band model is to construct protocols offering the optimal trade-off between their security and the length of their out-of-band authenticated value, and our protocols in this paper offer this optimal trade-off both in the user-to-user setting and in the group setting (within lower order terms). The proof of Theorem 4 may be found in the full version of this paper [45].

▶ **Theorem 4.** *Let $\ell = \ell(\lambda), n = n(\lambda)$ and $\epsilon = \epsilon(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. For any out-of-band (ℓ, ϵ)-key-exchange protocol (over any key space $\mathcal{K}$) for a group of size $n(\lambda)$, there exists a negligible function $\nu(\cdot)$ such that*

$$\epsilon(\lambda) \geq \min\left\{\frac{1}{3}, \frac{n(\lambda)-1}{4} \cdot 2^{-\ell}\right\} - \nu(\lambda)$$

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

Lazy users. Motivated by the recent work of Naor et al. [44], we consider in addition the security of out-of-band key-exchange protocols when executed by lazy users who may not consider the out-of-band value in its entirety (e.g., users who compare with each other only a subset of its positions). Given an out-of-band key-exchange protocol $\pi = \langle P_1, \ldots, P_n \rangle$ we define a collection of "lazy protocols", one per each possible subset of positions of the out-of-band authenticated value. Specifically, given a protocol π in which the out-of-band authenticated value consists of ℓ characters, for a subset $\mathcal{I} \subseteq [\ell]$ of indexes, we consider the "lazy protocol" $\pi_{\mathcal{I}}$ in which the parties execute π, with the exception that the party who sends the out-of-band value does not send the entire value, but rather sends only its substring that corresponds to the positions in the set $\mathcal{I}$ (we refer the reader to the work of Naor et al. [44] for an in-depth discussion of lazy protocols and of the motivation underlying them).

▶ **Definition 5.** *Let $n = n(\lambda), \ell = \ell(\lambda)$ and $\epsilon = \epsilon(\lambda, \cdot) : 2^{[\ell]} \to [0,1]$ be functions of the security parameter $\lambda \in \mathbb{N}$. A group out-of-band (ℓ, ϵ)-key-exchange protocol $\pi = \langle P_1, \ldots, P_n \rangle$ for a group of size $n(\lambda)$ is* secure for lazy users *if for every $\mathcal{I} = \mathcal{I}(\lambda) \subseteq [\ell]$ the lazy protocol $\pi_{\mathcal{I}}$ is a group out-of-band $(|\mathcal{I}|, \epsilon(\cdot, \mathcal{I}))$-key-exchange protocol for a group of size $n(\lambda)$.*

4 From Strong Authentication to Key Exchange

We show that any passively-secure key-exchange protocol can be transformed into an out-of-band authenticated key-exchange protocol that satisfies our strong notion of security (see Definition 3). Moreover, the resulting protocol offers the optimal trade-off between the length of its out-of-band value and its security within lower order terms (see Theorem 4). We prove the following theorem:

▶ **Theorem 6.** *Assuming the existence of any passively-secure key-exchange protocol, then for any functions $\ell = \ell(\lambda)$ and $n = n(\lambda)$ of the security parameter $\lambda \in \mathbb{N}$ there exists an (ℓ, ϵ)-out-of-band authenticated key-exchange protocol for a group of size $n(\lambda)$ over the same key space, where $\epsilon(\lambda) \leq 2 \cdot (n-1) \cdot (1/2 + o(1))^{\ell(\lambda)}$ for every $\lambda \in \mathbb{N}$.*

4.1 Immediate Message Delivery and Passively-Secure Immediate Key Delivery

Our construction relies on two main building blocks that satisfy a property similar to that of immediate key delivery, as defined in Section 3 via our experiment $\mathsf{FailStopExec}(\pi, A, \lambda)$ for modeling a fail-stop execution of a protocol (note that this experiment is well defined not only for key-exchange protocols, and can in fact be used to model aborting parties in a wide range of protocols).

Out-of-band message authentication with immediate message delivery. Our first building block is a strengthened form of an out-of-band group message authentication protocol, extending the notion introduced by Rotem and Segev [54] for such protocols (see Definition 2) by asking for *immediate message delivery*: Even if a subset of the parties aborts the execution of the authentication protocol before its completion, the remaining parties should still output the sender's input message m. Relying on the notion we introduced in Section 3, this property is formalized by strengthening Definition 2 as follows:

▶ **Definition 7.** *Let $\ell = \ell(\lambda), \epsilon = \epsilon(\lambda)$ and $n = n(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. We say that an (ℓ, ϵ)-out-of-band group message authentication protocol $\pi = \langle S, R_1, \ldots, R_{n-1} \rangle$ for groups of size n and message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ provides* immediate message delivery*, if for every $\lambda \in \mathbb{N}$, for every algorithm A, and for every input message $m \in \mathcal{M}_\lambda$ to S, it holds that*

$$\Pr_{(\mathit{AbortSet}, m_1, \ldots, m_n) \leftarrow \mathit{FailStopExec}(\pi, A, \lambda)} [\forall i \in [n] \setminus \mathit{AbortSet} : m_i = m] = 1.$$

In Section 5 we show that an out-of-band message authentication protocol with immediate message delivery can be constructed based on the existence of any a statistically-binding concurrent non-malleable commitment scheme (and thus based on any one-way function – see the full version [45] for specifics). Moreover, the protocol we construct offers the optimal tradeoff between the length of its out-of-band value and its security (i.e., the adversary's forgery probability).

Passively-secure key exchange with immediate key delivery. Our second building block is a passively-secure key-exchange protocol with immediate key delivery. This is naturally defined by replacing the standard correctness requirement of passively-secure key-exchange protocols with our immediate key delivery requirement stated in Definition 3. A passively-secure key exchange protocol $\langle P_1, \ldots, P_n \rangle$ with immediate key delivery can be easily obtained, for example, from any user-to-user passively-secure key-exchange protocol via the following simple transformation:

1. P_1 samples a random key $\mathsf{k} \leftarrow \mathcal{K}_\lambda$.
2. For every $i \in \{2, \ldots, n\}$, P_1 and P_i invoke the user-to-user key-exchange protocol and establish a shared key k_i.
3. For every $i \in \{2, \ldots, n\}$, P_1 uses a CPA-secure symmetric encryption scheme (whose existence is implied by that of any one-way function) to encrypt k using key k_i, and sends the resulting ciphertext to P_i.
4. Each P_i uses k_i from Step 2 to decrypt the received ciphertext, and then outputs the result of the decryption. P_1 outputs k.

It is straightforward to verify that this transformation indeed yields a passively-secure group key-exchange protocol with immediate key delivery: Even if a subset of the parties aborts the execution of the protocol before its completion, the remaining parties all output the key k chosen by P_1.

4.2 Our Construction

Our protocol relies on the following building blocks:

- A group (ℓ, ϵ)-out-of-band message authentication protocol $\langle S, R_1, \ldots, R_{n-1} \rangle$ with immediate message delivery, where $\ell = \ell(\lambda)$ and $\epsilon = \epsilon(\lambda)$ are functions of the security parameter $\lambda \in \mathbb{N}$.
- A passively-secure group key-exchange protocol $\langle P_{\mathsf{KE},1}, \ldots, P_{\mathsf{KE},n} \rangle$ with key space $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda \in \mathbb{N}}$ and immediate key delivery. We assume without loss of generality that each party in $\{P_2, \ldots, P_n\}$ sends messages to and receives messages from P_1 only.[10]
- A one-time strongly-unforgeable signature scheme $(\mathsf{KG}, \mathsf{Sign}, \mathsf{Vrfy})$.

Our protocol, which is denoted by $\langle P_1, \ldots, P_n \rangle$ and formally described below, starts by using the underlying out-of-band message authentication protocol for authenticating a verification key for the one-time signature scheme. This verification key is generated by the initiating party (denoted P_1), and its corresponding signing key is then used to sign the transcript of the out-of-band message authentication protocol, as well as the transcript of an execution of the passively-secure key-exchange protocol. The shared key resulting from executing the passively-secure key-exchange protocol is the output of each party, assuming that from this party's point of view the signature verifies correctly and the out-of-band message authentication protocol terminates successfully (i.e., no forgery was detected).

For describing the protocol, we assume for simplicity of presentation that all messages in the protocols $\langle S, R_1, \ldots, R_{n-1} \rangle$ and $\langle P_{\mathsf{KE},1}, \ldots, P_{\mathsf{KE},n} \rangle$ are sent to all participating users (and hence, the transcript of an honest execution of each of the protocols is the same according to the view of all users).

Out-of-Band Authenticated Group Key-Exchange Protocol $\langle P_1, \ldots, P_n \rangle$

Joint input: The security parameter 1^λ.

1. P_1 samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KG}(1^\lambda)$ and sends vk to all other users.
2. $P_1, \ldots, P_n$ execute the out-of-band message authentication protocol $\langle S, R_1, \ldots, R_{n-1} \rangle$, where P_1 runs S on input $(1^\lambda, \mathsf{vk})$, and P_i runs R_{i-1} on input 1^λ for every $i \in \{2, \ldots, n\}$. Denote by $\widehat{\mathsf{vk}}_i$ the output of R_{i-1} in this execution.
3. $P_1, \ldots, P_n$ execute the passively-secure key-exchange protocol $\langle P_{\mathsf{KE},1}, \ldots, P_{\mathsf{KE},n} \rangle$, where P_i runs $P_{\mathsf{KE},i}$ on input 1^λ for every $i \in [n]$. Denote by k_i the output of $P_{\mathsf{KE},i}$ in this execution.
4. Denote by trans_i the transcript of Steps 2 and 3 according to the view of P_i. P_1 computes $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{trans}_1)$ and sends σ to $P_2, \ldots, P_n$.
5. Denote by $\widehat{\sigma}_i$ the signature received by P_i for $i \in \{2, \ldots, n\}$. If $\widehat{\mathsf{vk}}_i \neq \perp$ and $\mathsf{Vrfy}(\widehat{\mathsf{vk}}_i, \mathsf{trans}_i, \widehat{\sigma}_i) = 1$ then P_i outputs k_i, and otherwise P_i outputs $\perp$.

The following theorem – which due to space limitations is proven in the full version of this paper [45] – establishes the correctness and security of our protocol according to Definition 3:

[10] Note that this is the case in the construction from any passively-secure (user-to-user) key-exchange protocol sketched in Section 4.1. Moreover, any passively-secure group key-exchange protocol can be easily compiled into one in which all parties communicate directly solely with P_1, by re-routing all messages through P_1 (i.e., if P_i wishes to send some message to P_j, it sends it to P_1 who then forwards it to P_j).

▶ **Theorem 8.** *Let $\ell = \ell(\lambda), \epsilon = \epsilon(\lambda)$ and $n = n(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. Then $\langle P_1, \ldots, P_n \rangle$ is an (ℓ, ϵ)-out-of-band authenticated group key-exchange protocol with key space $\mathcal{K}$ for groups of size n, assuming that:*

1. *$\langle P_{\mathsf{KE},1}, \ldots, P_{\mathsf{KE},n} \rangle$ is a passively-secure group key-exchange protocol with key space $\mathcal{K}$ and immediate key delivery.*
2. *$(\mathsf{KG}, \mathsf{Sign}, \mathsf{Vrfy})$ is a one-time strongly-unforgeable signature scheme.*
3. *$\langle S, R_1, \ldots, R_{n-1} \rangle$ is a group (ℓ, ϵ)-out-of-band message authentication protocol with immediate message delivery for $n - 1$ receivers.*

If, in addition, $\langle S, R_1, \ldots, R_{n-1} \rangle$ is secure when executed by lazy users, then $\langle P_1, \ldots, P_n \rangle$ is secure when executed by lazy users.

Note that the existence of any user-to-user passively-secure key-exchange protocol implies the existence of a one-way function, which in turn implies the existence of a strongly-unforgeable signature scheme, and (as we show in Section 5) of a group (ℓ, ϵ)-out-of-band message authentication protocol with immediate message delivery and $\epsilon(\lambda) \leq 2 \cdot n(\lambda) \cdot (1/2 + o(1))^{\ell(\lambda)}$. In addition, as discussed in Section 4.1, any user-to-user passively-secure key-exchange protocol implies the existence of such a protocol with immediate key delivery. Theorem 6 thus immediately follows as a corollary of Theorem 8.

5 Out-of-Band Message Authentication with Immediate Message Delivery

In this section we construct a group out-of-band message authentication protocol with immediate message delivery based on the existence of any one-way function (instantiating the required building blocks in the random-oracle model leads to a concrete and efficient protocol). We prove the following theorem:

▶ **Theorem 9.** *Let $\ell = \ell(\lambda)$ and $n = n(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$ and assume the existence of one-way functions. Then, there exists a group (ℓ, ϵ)-out-of-band message authentication protocol for $n(\lambda)$ receivers with immediate message delivery, where $\epsilon(\lambda) = 2 \cdot n(\lambda) \cdot (1/2 + o(1))^{\ell(\lambda)}$.*

For a string $s \in \{0, 1\}^*$ and an index $i \in [|s|]$, we let s_i (or $(s)_i$) denote the ith bit of s. Let $\mathsf{BitWiseMajority}$ be the operation that on input a set of strings $s^1, \ldots, s^q$ of length ℓ, returns a string s^* whose kth coordinate is the majority among the kth coordinates of $s^1, \ldots, s^q$; i.e., if $\mathsf{BitWiseMajority}(s^1, \ldots, s^q) = s^*$, then for every $k \in [\ell]$, $(s^*)_k = \mathsf{Majority}((s^1)_k, \ldots, (s^q)_k)$. Our protocol, denoted by π, is parameterized by the number of receivers $n = n(\lambda)$ and by a function $T = T(\lambda)$ of the security parameter $\lambda \in \mathbb{N}$. The protocol uses as a building block a statistically-binding concurrent non-malleable commitment scheme Com. As a commitment scheme may be interactive (unless one assumes the random-oracle model), when describing our protocol and referring to a commitment to a certain value, 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).

Group Out-of-Band Message Authentication Protocol $\pi = \langle S, R_1 \ldots, R_n \rangle$

Joint input: The security parameter 1^λ.

Phase 0: Initialization

1. Each party initializes a set of aborted receivers, based on her view of the protocol. We denote by $\mathcal{A}_S$ the set initialized by S, and by $\mathcal{A}_i$ the set initialized by each R_i. At the beginning of the execution $\mathcal{A}_S = \mathcal{A}_1 = \cdots = \mathcal{A}_n = \emptyset$.

Phase 1: Commitments for string selection

2. The sender S, on input m, chooses a random string $r_s \leftarrow \{0,1\}^\ell$, and executes n (possibly parallel) executions of Com to commit to the message (m, r_s) to each receiver R_i. Denote the resulting commitments according to the view of S by $c_s^1, \ldots, c_s^n$, and denote the commitment received by each R_i by $\widehat{c_s^i}$. S also appends to the first message it sends each R_i the message m. Denote by $\widehat{m}_i$ the message received by each R_i.
3. Each receiver R_i chooses random ℓ-bit strings $r_{i,1}, \ldots, r_{i,T} \leftarrow \{0,1\}^\ell$, and commits to them to the sender S using T (parallel) executions of Com. For every $i \in [n]$ denote the resulting commitments according to the view of R_i by $c_{i,1}, \ldots, c_{i,T}$, and denote the commitments received by S by $\widehat{c_{i,1}}, \ldots, \widehat{c_{i,T}}$. If some receiver R_i aborts during the commitment protocol, then S updates $\mathcal{A}_S = \mathcal{A}_S \cup \{i\}$.
4. For every $i \in [n]$, S forwards to R_i the commitments $\left\{\left(\widehat{c_{j,1}}, \ldots, \widehat{c_{j,T}}\right)\right\}_{j \in [n] \setminus \{i\}}$ received by her in Step 3 of the protocol, as well as $\mathcal{A}_S$. We denote by $\left\{(\widehat{c_{j,1 \to i}}, \ldots, \widehat{c_{j,T \to i}})\right\}_{j \in [n] \setminus \{i\}}$ and $\widehat{\mathcal{A}_S}_i$ the forwarded commitments and the aborted set, respectively, as received by R_i. In addition, R_i updates $\mathcal{A}_i = \widehat{\mathcal{A}_S}_i$.

Phase 2: Gradual decommitments for string selection

5. For $t = 1, \ldots, T$:
   - **a.** For every $i \in [n]$, R_i sends to S a decommitment $d_{i,t}$ of her commitment $c_{i,t}$ from Step 3. Let $\widehat{d_{i,t}}$ denote the decommitment received by S. For every $i \in [n]$ the sender S then checks whether $\widehat{d_{i,t}}$ is a valid decommitment to $\widehat{c_{i,t}}$. If so, let $\widehat{r_{i,t}}$ denote the committed value. If some receiver R_i either sends an invalid decommitment or aborts before sending $d_{i,t}$, then S updates $\mathcal{A}_S = \mathcal{A}_S \cup \{i\}$. For every $i \in \mathcal{A}_S$, S lets $\widehat{r_{i,t}} = 0^\ell$.
   - **b.** For every $i \in [n]$, S forwards R_i the decommitments $(\widehat{d_{j,t}})_{j \in [n] \setminus \{i\}}$, as well as $\mathcal{A}_S$. We let $(\widehat{d_{j,t \to i}})_{j \in [n] \setminus \{i\}}$ and $\widehat{\mathcal{A}_S}_i$ denote the decommitments and the set received by R_i, respectively. R_i updates $\mathcal{A}_i = \mathcal{A}_i \cup \widehat{\mathcal{A}_S}_i$. If for some $j \in [n] \setminus (\mathcal{A}_i \cup \{i\})$ it holds that $\widehat{d_{j,t \to i}}$ is not a valid decommitment to $\widehat{c_{j,t \to i}}$ received by R_i is Step 4, then R_i updates $\mathcal{A}_i = \mathcal{A}_i \cup \{j\}$. Otherwise, denote by $(\widehat{r_{j,t \to i}})_{j \in [n] \setminus \{i\}}$ the values obtained by opening the commitments. For every $j \in \mathcal{A}_i$, R_i lets $\widehat{r_{j,t \to i}} = 0^\ell$.
   - **c.** S computes $\sigma_t = \bigoplus_{i \in [n]} \widehat{r_{i,t}}$, and for every $i \in [n]$, R_i computes $\widehat{\sigma}_{i,t} = r_{i,t} \bigoplus_{j \in [n] \setminus \{i\}} \widehat{r_{j,t \to i}}$.
6. For every $i \in [n]$, the sender S sends receiver R_i a decommitment d_s^i to the corresponding commitment from Step 2. Denote by $\widehat{d_s^i}$ the decommitment received by R_i. For every $i \in [n]$ the receiver R_i checks if $\widehat{d_s^i}$ is a valid decommitment to $\widehat{c_s^i}$. If it is, denote the committed value by $(\widehat{m}_i, \widehat{r_s^i})$. If it is not a valid decommitment, then R_i outputs $\perp$ and terminates.

Phase 3: Out-of-band verification

7. S computes $\sigma_R = \mathsf{BitWiseMajority}(\sigma_1, \ldots, \sigma_T)$ and sends $\sigma = r_s \oplus \sigma_R$ over the out-of-band channel. For every $i \in [n]$, R_i computes $\widehat{\sigma_R}_i = \mathsf{BitWiseMajority}(\widehat{\sigma}_{i,1}, \ldots, \widehat{\sigma}_{i,T})$, and outputs $\widehat{m}_i$ if $\sigma = \widehat{r_s^i} \oplus \widehat{\sigma_R}_i$. Otherwise, R_i outputs $\perp$.

The following theorem captures the security of our protocol, and its proof can be found in the full version of this paper [45] (recall that in Section 1.3 we provided a high-level overview of the proof). Setting $T(\lambda) = (n(\lambda))^2 \cdot \omega(1)$ yields Theorem 9 as an immediate corollary.

▶ **Theorem 10.** *Let $\ell = \ell(\lambda), T = T(\lambda)$ and $n = n(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$, and let* Com *be a statistically-binding concurrent non-malleable commitment scheme. Then, the protocol π is a group (ℓ, ϵ)-out-of-band message authentication protocol for n receivers with immediate message delivery, where $\epsilon(\lambda) = 2 \cdot n(\lambda) \cdot \left(1/2 + O\left(n(\lambda)/\sqrt{T(\lambda)}\right)\right)^{\ell(\lambda)}$.*

References

1 Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In *Proceedings of the 8th International Conference on Practice and Theory in Public-Key Cryptography*, pages 65–84, 2005.

2 Joël Alwen, Sandro Coretti, and Yevgeniy Dodis. The double ratchet: Security notions, proofs, and modularization for the signal protocol. In *Advances in Cryptology – EUROCRYPT '19*, pages 129–158, 2019.

3 Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*. John Wiley & Sons, 2004.

4 Baruch Awerbuch, Manuel Blum, Benny Chor, Shafi Goldwasser, and Silvio Micali. How to implement Bracha's $O(\log n)$ byzantine agreement algorithm. Unpublished manuscript, 1985.

5 Richard Barnes, Jon Millican, Emad Omara, Katriel Cohn-Gordon, and Raphael Robert. The messaging layer security protocol, 2019. Available at `https://datatracker.ietf.org/doc/draft-ietf-mls-protocol/` (accessed 11-Dec-2019).

6 Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of the 30th annual ACM Symposium on Theory of Computing*, pages 419–428, 1998.

7 Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *Advances in Cryptology – EUROCRYPT '00*, pages 139–155, 2000.

8 Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In *Advances in Cryptology – CRYPTO '93*, pages 232–249, 1993.

9 Mihir Bellare and Phillip Rogaway. Provably secure session key distribution: the three party case. In *Proceedings of the 27th annual ACM Symposium on Theory of Computing*, pages 57–66, 1995.

10 Mihir Bellare, Asha Camper Singh, Joseph Jaeger, Maya Nyayapati, and Igors Stepanovs. Ratcheted encryption and key exchange: The security of messaging. In *Advances in Cryptology – CRYPTO '17*, pages 619–650, 2017.

11 Bluetooth Special Interest Group. Bluetooth core specification v. 5.1, 2019. Available at `https://www.bluetooth.com/specifications/bluetooth-core-specification/` (accessed 11-Dec-2019).

12 Victor Boyko, Philip MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Advances in Cryptology – EUROCRYPT '00*, pages 156–171, 2000.

13 Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Provably authenticated group Diffie-Hellman key exchange - the dynamic case. In *Advances in Cryptology – ASIACRYPT '01*, pages 290–309, 2001.

14 Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Dynamic group Diffie-Hellman key exchange under standard assumptions. In *Advances in Cryptology – EUROCRYPT '02*, pages 321–336, 2002.

15 Emmanuel Bresson, Olivier Chevassut, David Pointcheval, and Jean Jacques Quisquater. Provably authenticated group Diffie-Hellman key exchange. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 255–264, 2001.

16 Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology – EUROCRYPT '01*, pages 453–474, 2001.

17 Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four-round concurrent non-malleable commitments from one-way functions. In *Advances in Cryptology – CRYPTO '17*, pages 127–157, 2017.

18 Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 364–369, 1986.

19 Katriel Cohn-Gordon, Cas Cremers, Luke Garratt, Jon Millican, and Kevin Milner. On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees. In *Proceedings of the 25th ACM conference on Computer and Communications Security*, pages 1802–1819, 2018.

20 Katriel Cohn-Gordon, Cas J.F̃. Cremers, Benjamin Dowling, Luke Garratt, and Douglas 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.

21 Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

22 Alexis Duque. Deep dive into Bluetooth LE security. *Medium.* Available at `https://medium.com/rtone-iot-security/deep-dive-into-bluetooth-le-security-d2301d640bfc`, 2018.

23 F. Betül Durak and Serge Vaudenay. Bidirectional asynchronous ratcheted key agreement without key-update primitives. In *Advances in Information and Computer Security – IWSEC '19*, pages 343–362, 2019.

24 Dario Fiore, Maria Isabel Gonzalez Vasco, and Claudio Soriente. Partitioned group password-based authenticated key exchange. *The Computer Journal*, 60(12):1912–1922, 2017.

25 Tilman Frosch, Christian Mainka, Christoph Bader, Florian Bergsma, Jörg Schwenk, and Thorsten Holz. How secure is TextSecure? In *Proceedings of the 1st IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 457–472, 2016.

26 Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In *Advances in Cryptology – EUROCRYPT '03*, pages 524–543, 2003.

27 Michael T. Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. Loud and clear: Human-verifiable authentication based on audio. In *26th IEEE International Conference on Distributed Computing Systems*, page 10, 2006.

28 Vipul 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.

29 David Jablon. Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, 26(5):5–26, 1996.

30 Joseph Jaeger and Igors Stepanovs. Optimal channel security against fine-grained state compromise: The safety of messaging. In *Advances in Cryptology – CRYPTO '18*, pages 33–62, 2018.

31 Daniel Jost, Ueli Maurer, and Marta Mularczyk. Efficient ratcheting: Almost-optimal guarantees for secure messaging. In *Advances in Cryptology – EUROCRYPT '19*, pages 159–188, 2019.

32 Ronald Kainda, Ivan Flechais, and AW Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *Symposium on usable privacy and security (SOUPS)*, pages 11:1–11:12, 2009.

33 Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *Advances in Cryptology – EUROCRYPT '01*, pages 475–494, 2001.

34 Jonathan Katz and Moti Yung. Scalable protocols for authenticated group key exchange. In *Advances in Cryptology – CRYPTO '03*, pages 110–125, 2003.

35 Nadim Kobeissi, Karthikeyan Bhargavan, and Bruno 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.

36 Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In *International Conference on Provable Security '07*, pages 1–16, 2007.

37 Sampsa Latvala, Mohit Sethi, and Tuomas Aura. Evaluation of out-of-band channels for IoT security. *SN Computer Science*, 1(1):1–18, 2019.

38 Sven Laur and Kaisa Nyberg. Efficient mutual data authentication using manually authenticated strings. In *International Conference on Cryptology and Network Security*, pages 90–107, 2006.

39 Huijia Lin and Rafael 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.

40 Yehuda Lindell. Comparison-based key exchange and the security of the numeric comparison mode in bluetooth v2.1. In *CT-RSA '09*, pages 66–83, 2009.

41 Nancy A. Lynch. *Distributed algorithms.* Elsevier, 1996.

42 Rene Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In *International Conference on Pervasive Computing*, pages 144–161, 2007.

43 Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, pages 110–124, 2005.

44 Moni Naor, Lior Rotem, and Gil Segev. The security of lazy users in out-of-band authentication. In *Proceedings of the 16th Theory of Cryptography Conference*, pages 575–599, 2018.

45 Moni Naor, Lior Rotem, and Gil Segev. Out-of-band authenticated group key exchange: From strong authentication to immediate key delivery. Cryptology ePrint Archive, Report 2019/1458, 2019.

46 Moni Naor, Gil Segev, and Adam 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.

47 Moni Naor, Gil Segev, and Adam 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.

48 Sylvain Pasini and Serge Vaudenay. An optimal non-interactive message authentication protocol. In *CT-RSA '06*, pages 280–294, 2006.

49 Sylvain Pasini and Serge Vaudenay. SAS-based authenticated key agreement. In *Proceedings on the 9th International Conference on Theory and Practice of Public-Key Cryptography*, pages 395–409, 2006.

50 Trevor Perrin and Moxie Marlinspike. The double ratchet algorithm, 2016. Available at `https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf` (accessed 11-Dec-2019).

51 Bertram Poettering and Paul Rösler. Towards bidirectional ratcheted key exchange. In *Advances in Cryptology – CRYPTO '18*, pages 3–32, 2018.

52 Bertram Poettering and Paul Rösler. Asynchronous ratcheted key exchange. Cryptology ePrint Archive, Report 2018/296, 2018.

53 Ronald L. Rivest and Adi Shamir. How to expose an eavesdropper. *Communications of the ACM*, 27(4):393–395, 1984.

54 Lior Rotem and Gil Segev. Out-of-band authentication in group messaging: Computational, statistical, optimal. In *Advances in Cryptology – CRYPTO '18*, pages 63–89, 2018.

55 Lior Rotem and Gil Segev. Out-of-band authentication in group messaging: Computational, statistical, optimal. Cryptology ePrint Archive, Report 2018/493, 2018.

56 Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *IEEE Symposium on Security and Privacy*, pages 306–313, 2006.

57 Michael Schliep and Nicholas Hopper. End-to-end secure mobile group messaging with conversation integrity and deniability. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 55–73, 2019.

58 Victor Shoup. On formal models for secure key exchange. Theory of Cryptography Library (available at `www.shoup.net/papers/skey.pdf`), 1999.

59 Telegram. End-to-end encrypted voice calls – key verification. Available at `https://core.telegram.org/api/end-to-end/voice-calls#key-verification` (accessed 11-Dec-2019).

60 Telegram. End-to-end encryption. Available at `https://core.telegram.org/api/end-to-end` (accessed 11-Dec-2019).

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

62 Viber encryption overview. Available at `https://www.viber.com/app/uploads/Viber-Encryption-Overview.pdf`.

63 WhatsApp encryption overview. Available at `https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf`.