Tight Bounds for Unconditional Authentication Protocols in the Manual Channel and Shared Key Models*

Moni Naor**, Gil Segev, and Adam Smith***

Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot 76100, Israel.
{moni.naor, gil.segev, adam.smith}@weizmann.ac.il

Abstract. We address the message authentication problem in two seemingly different communication models. In the first model, the sender and receiver are connected by an insecure channel and by a low-bandwidth auxiliary channel, that enables the sender to "manually" authenticate one short message to the receiver (for example, by typing a short string or comparing two short strings). We consider this model in a setting where no computational assumptions are made, and prove that for any $0 < \epsilon < 1$ there exists a $\log^* n$-round protocol for authenticating n-bit messages, in which only $2\log(1/\epsilon) + \mathrm{O}(1)$ bits are manually authenticated, and any adversary (even computationally unbounded) has probability of at most ϵ to cheat the receiver into accepting a fraudulent message. Moreover, we develop a proof technique showing that our protocol is essentially optimal by providing a lower bound of $2\log(1/\epsilon) - 6$ on the required length of the manually authenticated string.

The second model we consider is the traditional message authentication model. In this model the sender and the receiver share a short secret key; however, they are connected only by an insecure channel. Once again, we apply our proof technique, and prove a lower bound of $2\log(1/\epsilon) - 2$ on the required Shannon entropy of the shared key. This settles an open question posed by Gemmell and Naor (CRYPTO '93).

Finally, we prove that one-way functions are *essential* (and sufficient) for the existence of protocols breaking the above lower bounds in the computational setting.

1 Introduction

Message authentication is one of the major issues in cryptography. Protocols for message authentication provide assurance to the receiver of a message that it was sent by a specified legitimate sender, even in the presence of an adversary who

* A longer version, including proofs of all claims, appears as [15].

** Incumbent of the Judith Kleeman Professorial Chair. Research supported in part by a grant from the Israel Science Foundation.

*** Research supported by the Louis L. and Anita M. Perlman postdoctoral fellowship.

controls the communication channel. For more than three decades, numerous authentication models have been investigated, and many authentication protocols have been suggested. The security of these protocols can be classified according to the assumed computational resources of the adversary. Security that holds when one assumes a suitable restriction on the adversary's computing capabilities is called *computational security*, while security that holds even when the adversary is computationally unbounded is called *unconditional security* or *information-theoretic security*. This paper is concerned mostly with unconditional security of a single instance of message authentication protocols. We remark that there are three main advantages to unconditional security over computational security. The first is the obvious fact that no assumptions are made about the adversary's computing capabilities or about the computational hardness of specific problems. The second, less apparent advantage, is that unconditionally secure protocols are often more efficient than computationally secure protocols. The third advantage is that unconditional security allows exact evaluation of the error probabilities.

Shared key authentication. The first construction of an authentication protocol in the literature was suggested by Gilbert, MacWilliams and Sloane [10] in the information-theoretic adversarial setting. They considered a communication model in which the sender and the receiver share a key, which is not known to the adversary. Gilbert et al. presented a non-interactive protocol, in which the length of the shared key is $2\max\{n, \log(1/\epsilon)\}$; henceforth, n is the length of the input message and ϵ is the adversary's probability of cheating the receiver into accepting a fraudulent message. They also proved a lower bound of $2\log(1/\epsilon)$ on the required entropy of the shared key in non-interactive deterministic protocols. Clearly, a trivial lower bound on this entropy is $\log(1/\epsilon)$, since an adversary can merely guess the shared key. This model, to which we refer as the *shared key model*, became the standard model for message authentication protocols. Protocols in this model should provide authenticity of messages while minimizing the length of the shared key.

Wegman and Carter [20] suggested using ϵ-*almost strongly universal*$_2$ hash functions for authentication. This enabled them to construct a non-interactive protocol in which the length of the shared key is $\mathrm{O}(\log n \log(1/\epsilon))$ bits. In 1984, Simmons [17] initiated a line of work on unconditionally secure authentication protocols (see, for example, [13, 18] and more references in the full version). Gemmell and Naor [9] proposed a non-interactive protocol, in which the length of the shared key is only $\log n + 5\log(1/\epsilon)$ bits. They also demonstrated that by introducing interaction, the length of the shared key can be made independent of the length of the input message. More specifically, they suggested a $\log^* n$-round protocol that enables the sender to authenticate n-bit messages, where the length of the shared key is only $2\log(1/\epsilon) + \mathrm{O}(1)$ bits. However, it was not known whether this upper bound is optimal, that is, if by introducing interaction the entropy of the shared key can be made smaller than $2\log(1/\epsilon)$.

Manual authentication. Recently, Vaudenay [19] formalized a realistic communication model for message authentication, in which the sender and the receiver are connected by a bidirectional insecure channel, and by a unidirectional low-bandwidth auxiliary channel, but do not share any secret information. It is assumed that the adversary has full control over the insecure channel. In particular, the adversary can read any message sent over this channel, prevent it from being delivered, and insert a new message at any point in time. The low-bandwidth auxiliary channel enables the sender to "manually" authenticate one short string to the receiver. The adversary cannot modify this short string. However, the adversary can still read it, delay it, and remove it. We refer to the auxiliary channel as the *manual channel*, and to this communication model as the *manual channel model*. Protocols in this model should provide authenticity of long messages[1] while minimizing the length of the manually authenticated string. We remark that $\log(1/\epsilon)$ is an obvious lower bound in this model as well.

The manual channel model is becoming very popular in real-world scenarios, whenever there are ad hoc networks with no trusted infrastructure. In particular, this model was found suitable for initial pairing of devices in wireless networks, such as Wireless USB [3] and Bluetooth[2] [2]. While in wired connections when a device is plugged in (i.e., when the wire is connected), the user can see that the connection is made, wireless connections may establish connection paths that are not straightforward. In fact, it may not be obvious when a device is connected or who its host is. Therefore, initial authentication in device and host connections is required so that the user will be able to validate both the device and its host.

Consider, for example, a user who wishes to connect a new DVD player to her home wireless network. Then, the user reading a short message from the display of the DVD player and typing it on a PC's keyboard constitutes a manual authentication channel from the DVD player to the PC. An equivalent channel is the user comparing two short strings displayed by the two devices, as suggested by Gehrmann et al. [8].

Constants do matter. The most significant constraint in the manual channel model is the length of the manually authenticated string. This quantity is determined by the environment in which the protocol is executed, and in particular by the capabilities of the user. While it is reasonable to expect a user to manually authenticate 20 or 40 bits, it is not reasonable to expect a user to manually authenticate 160 bits. Therefore, there is a considerable difference between manually authenticating $\log(1/\epsilon)$ or $2\log(1/\epsilon)$ bits, and manually authenticating a significantly longer string. This motivates the study of determining the exact lower bound on the required length of the manually authenticated string.

Our contribution. We present an unconditionally secure authentication protocol in the manual channel model, in which the sender manually authenticates only $2\log(1/\epsilon) + O(1)$ bits. Moreover, we develop a proof technique, proving that our protocol is essentially optimal in minimizing the length of the manually authenticated string. Then, we apply this technique to the shared key model, and settle an open question posed by Gemmell and Naor [9] by deriving a similar lower bound on the required entropy of the shared key. This lower bound

[1] Short messages can be directly manually authenticated.

[2] However, in existing protocols for pairing of Bluetooth devices, the manual channel is assumed to provide secrecy as well.

matches the upper bound of Gemmell and Naor. Finally, we consider these two communication models in the computational setting, and prove that one-way functions are essential for the existence of protocols breaking the above lower bounds.

Paper organization. The rest of the paper is organized as follows. We first briefly present some known definitions in Section 2. In Section 3 we describe the communication and adversarial models we deal with. Then, in Section 4 we present an overview of our results, and compare them to previous work. In Section 5 we propose an unconditionally secure message authentication protocol in the manual channel model. In Section 6 we describe the proof technique, that is then used to establish the optimality of our protocol. In Section 7 we apply the same proof technique to the shared key model, and prove a lower bound on the required entropy of the shared key. Finally, in Section 8 we prove that in the computational setting, one-way functions are essential for the existence of protocols breaking the above lower bounds.

2 Preliminaries

We first present some fundamental definitions from Information Theory. Then, we briefly present the definitions of *one-way* functions and *distributionally one-way* functions. All logarithms in this paper are to the base of 2. Let X, Y and Z denote random variables.

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

Definition 1. *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called* one-way *if it is computable in polynomial-time, and for every probabilistic polynomial-time Turing machine*[3] *$\mathcal{M}$, every polynomial p, and all sufficiently large n,*

$$\Pr\left[\mathcal{M}(f(x), 1^n) \in f^{-1}(f(x))\right] < \frac{1}{p(n)} \ ,$$

where the probability is taken uniformly over all the possible choices of $x \in \{0,1\}^n$ and all the possible outcomes of the internal coin tosses of $\mathcal{M}$.

Definition 2. *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called* distributionally one-way *if it is computable in polynomial-time, and there exists a constant $c > 0$ such that for every probabilistic polynomial-time Turing machine $\mathcal{M}$, the distribution defined by $x \circ f(x)$ and the distribution defined by $\mathcal{M}(f(x)) \circ f(x)$ are n^{-c}-statistically far*[4] *when $x \in_{\mathrm{R}} \{0,1\}^n$.*

[3] We note that uniformity is not essential to our results.

[4] The *statistical distance* between two distributions $\mathcal{D}$ and $\mathcal{F}$ is defined as $\Delta(\mathcal{D},\mathcal{F}) = \frac{1}{2}\sum_\alpha \left|\Pr_{x\leftarrow\mathcal{D}}[x = \alpha] - \Pr_{x\leftarrow\mathcal{F}}[x = \alpha]\right|$. The distributions $\mathcal{D}$ and $\mathcal{F}$ are said to be *ϵ-statistically far* if $\Delta(\mathcal{D},\mathcal{F}) \geq \epsilon$. Otherwise, $\mathcal{D}$ and $\mathcal{F}$ are *ϵ-statistically close.*

Informally, it is hard to find a random inverse of a distributionally one-way function, although finding some inverse may be easy. Clearly, any one-way function is also a distributionally one-way function, but the converse may not always be true. However, Impagliazzo and Luby [11] proved that the *existence* of both primitives is equivalent.

3 Communication and Adversarial Models

We consider the message authentication problem in a setting where the sender and the receiver are connected by a bidirectional insecure communication channel, over which an adversary has full control. In particular, the adversary can read any message sent over this channel, delay it, prevent it from being delivered, and insert a new message at any point in time.

3.1 The Manual Channel Communication Model

In addition to the insecure channel, we assume that there is a unidirectional low-bandwidth auxiliary channel, that enables the sender to "manually" authenticate one short string to the receiver. The adversary cannot modify this short string. However, the adversary can still read it, delay it, and remove it.

The input of the sender $\mathcal{S}$ in this model is a message m, which she wishes to authenticate to the receiver $\mathcal{R}$. The input message m can be determined by the adversary $\mathcal{A}$. In the first round, $\mathcal{S}$ sends the message m and an authentication tag x_1 over the insecure channel. In the following rounds only a tag x_i is sent over the insecure channel. The adversary receives each of these tags x_i and can replace them with $\widehat{x}_i$ of her choice, as well as replace the input message m with a different message $\widehat{m}$. In the last round, $\mathcal{S}$ may manually authenticate a short string s.

Notice that in the presence of a computationally unbounded adversary, additional insecure rounds (after the manually authenticated string has been sent) do not contribute to the security of the protocol. This is due to the fact that after reading the manually authenticated string, the unbounded adversary can always simulate the sender successfully (since the sender and the receiver do not share any secret information, and since the adversary has full control over the communication channel from this point on). Therefore, there is no loss of generality in assuming that the manually authenticated string is sent in the last round. This is true also in the computational setting, under the assumption that distributionally one-way functions do not exist. A generic protocol in this model is described in Figure 1.

We also allow the adversary to control the synchronization of the protocol's execution. More specifically, the adversary can carry on two separate, possibly asynchronous conversations, one with the sender and one with the receiver. However, the party that is supposed to send a message waits until it receives the adversary's message from the previous round.

When the input message m is chosen uniformly at random, the honest execution of the protocol defines a probability distribution on the message m, the tags x_i and the manually authenticated string s. We denote by M, X_i and S the random variables corresponding to m, x_i and s, respectively.

Definition 3. *An* unconditionally secure (n, ℓ, k, ϵ)-authentication protocol *in the manual channel model is a k-round protocol in the communication model described above, in which the sender wishes to authenticate an n-bit input message to the receiver, while manually authenticating at most ℓ bits. The following requirements must hold:*

1. **Completeness:** *For all input messages m, when there is no interference by the adversary in the execution, the receiver accepts m with probability at least* $1/2$.
2. **Unforgeability:** *For any computationally unbounded adversary, and for all input messages m, if the adversary replaces m with a different message $\widehat{m}$, then the receiver accepts $\widehat{m}$ with probability at most ϵ.*

In order to define the notion of a *computationally secure* authentication protocol, we actually consider a sequence of protocols by adding a security parameter t that defines the power of the adversaries against which each protocol in the sequence is secure. The completeness requirement is as in Definition 3. However, the unforgeability requirement now holds only against adversaries running in time $\mathrm{poly}(t)$, and we allow forgery probability of $\epsilon + \mathrm{negl}(t)$ for sufficiently large t. We refer the reader to the full version for the formal definition.

An authentication protocol in the manual channel model is said to be *perfectly complete* if for all input messages m, whenever there is no interference by the adversary in the execution, the receiver accepts m with probability 1.

3.2 The Shared Key Communication Model

In this model we assume that the sender and the receiver share a secret key s; however, they are connected only by an insecure channel. This key is not known to the adversary, but it is chosen from a probability distribution which is known to the adversary (usually the uniform distribution).

The input of the sender $\mathcal{S}$ in this model is a message m, which she wishes to authenticate to the receiver $\mathcal{R}$. The input message m can be determined by the adversary $\mathcal{A}$. In the first round, $\mathcal{S}$ sends the message m and an authentication tag x_1 over the insecure channel. In the following rounds only a tag x_i is sent over the insecure channel. The adversary receives each of these tags x_i and can replace them with $\widehat{x}_i$ of her choice, as well as replace the input message m with a different message $\widehat{m}$.

As in the manual channel model, in an honest execution we denote by S, M and X_i the random variables corresponding to s, m and x_i, respectively. A generic protocol in this model is described in Figure 1. As in the manual channel model, we allow the adversary to control the synchronization of the protocol's execution.

Fig. 1. Generic protocols in the manual channel model (left figure) and in the shared key model (right figure).

Definition 4. *An* unconditionally secure (n, ℓ, k, ϵ)-authentication protocol *in the shared key model is a k-round protocol in the communication model described above, in which the sender and the receiver share an ℓ-bit secret key, and the sender wishes to authenticate an n-bit input message to the receiver. The following requirements must hold:*

1. **Completeness:** *For all input messages m, when there is no interference by the adversary in the execution, the receiver accepts m with probability at least* $1/2$.
2. **Unforgeability:** *For any computationally unbounded adversary, and for all input messages m, if the adversary replaces m with a different message $\widehat{m}$, then the receiver accepts $\widehat{m}$ with probability at most ϵ.*

Similarly to the definitions in the manual channel model, we refer the reader to the full version for the definitions of a *computationally secure sequence* of authentication protocols and of a *perfectly complete* protocol.

4 Overview of Our Results and Comparison with Previous Work

Vaudenay [19] formalized the manual channel model, and suggested an authentication protocol in this model. Given $0 < \epsilon < 1$, Vaudenay's protocol enables the sender to authenticate an arbitrary long message to the receiver in three rounds, by manually authenticating $\log(1/\epsilon)$ bits. This protocol guarantees that, under the assumption that a certain type of non-interactive commitment scheme exists, the forgery probability of any polynomial-time adversary is at most $\epsilon + \nu(t)$, where $\nu(\cdot)$ is a negligible function and t is a security parameter. In particular, Laur, Asokan and Nyberg [12] proved that the required assumption is the existence of a non-interactive non-malleable commitment scheme. Dolev, Dwork and Naor [6] showed how to construct an *interactive* non-malleable commitment scheme from any one-way function, and therefore we obtain the following corollary:

Corollary 5 ([6, 12, 19]). *If one-way functions exist, then there exists a computationally secure $(n, \ell, k, \epsilon, t)$-authentication protocol in the manual channel model, with $t = poly(n, \ell, k)$ and $\ell = \log(1/\epsilon)$.*

However, the non-malleable commitment scheme suggested by Dolev, Dwork and Naor is inefficient, as it utilizes generic zero-knowledge proofs and its number of rounds is logarithmic in its security parameter. Therefore, the protocol implied by Corollary 5 is currently not practical (this is also true if the protocols in [1, 16] are used). Currently, the only known constructions of efficient non-malleable commitment schemes are in the random oracle model, or in the common random string model (see, for example, [4, 5]). These are problematic for the manual channel model, since they require a trusted infrastructure. This state of affairs motivates the study of a protocol that can be proved secure under more relaxed computational assumptions or even without any computational assumptions.

In Section 5, we present an unconditionally secure perfectly complete authentication protocol in the manual channel model. For any odd integer $k \geq 3$, and any integer n and $0 < \epsilon < 1$, our k-round protocol enables the sender to authenticate an n-bit input message to the receiver, while manually authenticating at most $2\log(1/\epsilon) + 2\log^{(k-1)} n + \mathrm{O}(1)$ bits. We prove that any adversary (even computationally unbounded) has probability of at most ϵ to cheat the receiver into accepting a fraudulent message. We note that our protocol only uses evaluations of polynomials over finite fields, for which very efficient implementations exist, and therefore it is very efficient and can be implemented on low-power devices. We prove the following theorem and corollary:

Theorem 6. *For any odd integer $k \geq 3$, and any integer n and $0 < \epsilon < 1$, there exists an unconditionally secure perfectly complete $(n, \ell = 2\log(1/\epsilon) + 2\log^{(k-1)} n + \mathrm{O}(1), k, \epsilon)$-authentication protocol in the manual channel model.*

Corollary 7. *For any integer n and $0 < \epsilon < 1$, the following unconditionally secure perfectly complete protocols exist in the manual channel model:*

1. *A $\log^* n$-round protocol in which at most $2\log(1/\epsilon) + \mathrm{O}(1)$ bits are manually authenticated.*
2. *A 3-round protocol in which at most $2\log(1/\epsilon) + \log\log n + \mathrm{O}(1)$ bits are manually authenticated.*

In Section 6, we develop a proof technique for deriving lower bounds on unconditionally secure authentication protocols, which allows us to show that our $\log^* n$-round protocol is optimal with respect to the length of the manually authenticated string. Specifically, we prove the following theorem:

Theorem 8. *For any unconditionally secure (n, ℓ, k, ϵ)-authentication protocol in the manual channel model, it holds that if $n \geq 2\log(1/\epsilon) + 4$, then $\ell > 2\log(1/\epsilon) - 6$.*

In Section 7 we consider the shared key communication model. Intensive research has been devoted to proving lower bounds on the required entropy of the shared key in unconditionally secure protocols. It was proved in several papers (see, for example, [13]), that in any perfectly complete non-interactive protocol, the required entropy of the shared key is at least $2\log(1/\epsilon)$. In addition, for such protocols, Gemmell and Naor [9] proved a lower bound of $\log n + \log(1/\epsilon) - \log\log(n/\epsilon) - 2$. Thus, there does not exist a perfectly complete non-interactive protocol that achieves the $2\log(1/\epsilon)$ bound. However, Gemmell and Naor also presented an *interactive* protocol that achieves the $2\log(1/\epsilon)$ bound. We remark that it was not previously known whether this bound is optimal for interactive protocols. By applying the proof technique described in Section 6, we settle this long-standing open question, proving the optimality of the protocol suggested by Gemmell and Naor.

Theorem 9. *For any unconditionally secure (n, ℓ, k, ϵ)-authentication protocol in the shared key model, it holds that $\mathrm{H}(S) \geq 2\log(1/\epsilon) - 2$, where S is the ℓ-bit shared key.*

Theorems 8 and 9 indicate that the two corresponding communication models are not equivalent: While in the manual channel model a lower bound can hold only when $n \geq \log(1/\epsilon)$, in the shared key model the lower bound holds even when authenticating only one bit. Nevertheless, the technique we develop applies to both models.

The idea underlying the lower bound proofs for the communication models under consideration can be briefly summarized as follows. First, we represent the entropies of the manually authenticated string and of the shared key by splitting them in a way that captures their reduction during the execution of the protocol. This representation allows us to prove that both the sender and the receiver must each independently reduce the entropies by at least $\log(1/\epsilon)$ bits. This is proved by considering two possible natural attacks on the given protocol. In these attacks we use the fact that the adversary is computationally unbounded in that she can sample distributions induced by the protocol. This usage of the adversary's capabilities, can alternatively be seen as randomly inverting functions given the image of a random input.

In Section 8, we take advantage of this point of view and prove that one-way functions are essential for the existence of protocols breaking the above lower bounds in the computational setting. Specifically, we show that if distributionally one-way functions do not exist, then a polynomial-time adversary can run the above mentioned attacks with almost the same success probability. The following theorem is proved (the reader is referred to the full version for a similar statement in the shared key model):

Theorem 10. *In the manual channel model, if there exists a computationally secure $(n, \ell, k, \epsilon, t)$-authentication protocol, such that $n \geq 2\log(1/\epsilon) + 4$, $\ell < 2\log(1/\epsilon) - 8$ and $t = \Omega(poly(n, k, 1/\epsilon))$, then one-way functions exist.*

A similar flavor of statement has recently been proved by Naor and Rothblum [14] in the context of memory checking, showing that one-way functions are essential for efficient on-line memory checking. Both results are based on combinatorial constructions (in our case these are the two attacks carried by an unbounded adversary), which are shown to be polynomial-time computable if one-way functions do not exist. However, we note that whereas Naor and Rothblum obtained asymptotic results (there is a multiplicative constant between the upper bound and the lower bound), we detect a sharp threshold.

5 The Message Authentication Protocol

In this section we prove Theorem 6 and Corollary 7 by constructing an authentication protocol, P_k. The protocol is based on the hashing technique suggested by Gemmell and Naor [9], in which the two parties reduce in each round the problem of authenticating the original message to that of authenticating a shorter message. In the first round the input message is sent, and then in each round the two parties *cooperatively* choose a hash function that defines a small, random "fingerprint" of the input message that the receiver should have received. If the adversary has changed the input message, then with high probability the fingerprint for the message received by the receiver will not match the fingerprint for the message that was sent by the sender. In a preliminary version of [9], this hashing technique was susceptible to synchronization attacks, as noted by Gehrmann [7]. However, in the full version of their paper, this was corrected by making *both* parties choose the random hash function used for fingerprinting the message.

We improve the hashing technique suggested by Gemmell and Naor as follows. First, we apply a different hash function, which enables us to manually authenticate a shorter string. A direct adaptation of the original hash function to the manual channel model would require the sender to manually authenticate at least $3\log(1/\epsilon)$ bits, while our construction manages to reduce this amount to only $2\log(1/\epsilon) + \mathrm{O}(1)$ bits. In addition, our protocol is asymmetric in the following sense: The roles of the sender and the receiver in cooperatively choosing the hash function are switched in every round. This enables us to deal with the fact that the adversary can read and delay any manually authenticated string.

Preliminaries. Denote by $\mathrm{GF}[Q]$ the Galois field with Q elements. For a message $m = m_1 \dots m_k \in \mathrm{GF}[Q]^k$ and $x \in \mathrm{GF}[Q]$ let $C_x(m) = \sum_{i=1}^{k} m_i x^i$. In other words, m is parsed as a polynomial of degree k over $\mathrm{GF}[Q]$ (without a constant term), and evaluated at the point x. Then, for any two different messages $m, \widehat{m} \in \mathrm{GF}[Q]^k$ and for any $c, \widehat{c} \in \mathrm{GF}[Q]$ the polynomials $C_x(m) + c$ and $C_x(\widehat{m}) + \widehat{c}$ are different as well, and therefore $\Pr_{x \in_{\mathrm{R}} \mathrm{GF}[Q]}\left[C_x(m) + c = C_x(\widehat{m}) + \widehat{c}\right] \leq \frac{k}{Q}$. We will use $C(\cdot)$ as a hash function to reduce the length of the message.

The construction. In protocol P_k we apply a sequence of hash functions $C^1, \dots, C^{k-1}$ in order to obtain a shorter and shorter message. Specifically, given the length, n, of the input message and the upper bound, ϵ, on the adversary's forgery probability, each C^j parses n_j-bit strings to polynomials over $\mathrm{GF}[Q_j]$, where $n_1 = n$, $\frac{2^{k-j} n_j}{\epsilon} \leq Q_j < \frac{2^{k-j+1} n_j}{\epsilon}$, and $n_{j+1} = \lceil 2 \log Q_j \rceil$. The protocol is described in Figure 2. Since the adversary can replace any authentication tag sent by any one of the parties over the insecure channel, then for such a tag x we denote by $\widehat{x}$ the tag that was actually received by the other party. Note that addition and multiplication are defined by the $\mathrm{GF}[Q_j]$ structures, and that $\langle u, v\rangle$ denotes the concatenation of the strings u and v.

Protocol $\mathbf{P}_k$:

1. $\mathcal{S}$ sends $m_{\mathcal{S}}^1 = m$ to $\mathcal{R}$.
2. $\mathcal{R}$ receives $m_{\mathcal{R}}^1$.
3. For $j = 1$ to $k-1$:
   (a) If j is odd, then
      i. $\mathcal{S}$ chooses $i_{\mathcal{S}}^j \in_{\mathrm{R}} \mathrm{GF}[Q_j]$ and sends it to $\mathcal{R}$.
      ii. $\mathcal{R}$ receives $\widehat{i_{\mathcal{S}}^j}$, chooses $i_{\mathcal{R}}^j \in_{\mathrm{R}} \mathrm{GF}[Q_j]$, and sends it to $\mathcal{S}$.
      iii. $\mathcal{S}$ receives $\widehat{i_{\mathcal{R}}^j}$, and computes $m_{\mathcal{S}}^{j+1} = \langle \widehat{i_{\mathcal{R}}^j}, C^j_{\widehat{i_{\mathcal{R}}^j}}(m_{\mathcal{S}}^j) + i_{\mathcal{S}}^j \rangle$.
      iv. $\mathcal{R}$ computes $m_{\mathcal{R}}^{j+1} = \langle i_{\mathcal{R}}^j, C^j_{i_{\mathcal{R}}^j}(m_{\mathcal{R}}^j) + \widehat{i_{\mathcal{S}}^j} \rangle$.
   (b) If j is even, then
      i. $\mathcal{R}$ chooses $i_{\mathcal{R}}^j \in_{\mathrm{R}} \mathrm{GF}[Q_j]$ and sends it to $\mathcal{S}$.
      ii. $\mathcal{S}$ receives $\widehat{i_{\mathcal{R}}^j}$, chooses $i_{\mathcal{S}}^j \in_{\mathrm{R}} \mathrm{GF}[Q_j]$, and sends it to $\mathcal{R}$.
      iii. $\mathcal{R}$ receives $\widehat{i_{\mathcal{S}}^j}$, and computes $m_{\mathcal{R}}^{j+1} = \langle \widehat{i_{\mathcal{S}}^j}, C^j_{\widehat{i_{\mathcal{S}}^j}}(m_{\mathcal{R}}^j) + i_{\mathcal{R}}^j \rangle$.
      iv. $\mathcal{S}$ computes $m_{\mathcal{S}}^{j+1} = \langle i_{\mathcal{S}}^j, C^j_{i_{\mathcal{S}}^j}(m_{\mathcal{S}}^j) + \widehat{i_{\mathcal{R}}^j} \rangle$.
4. $\mathcal{S}$ manually authenticates $m_{\mathcal{S}}^k$ to $\mathcal{R}$.
5. $\mathcal{R}$ accepts if and only if $m_{\mathcal{S}}^k = m_{\mathcal{R}}^k$.

Fig. 2. The k-round authentication protocol.

Note that the two parties can combine some of their messages, and therefore the protocol requires only k rounds of communication. An alternative way to describe the protocol is in a recursive fashion. The k-round protocol consists of $\mathcal{S}$ sending the message $m_1 = m$, as well as $\mathcal{S}$ and $\mathcal{R}$ exchanging $i_{\mathcal{S}}^1$, and $i_{\mathcal{R}}^1$. Then the two parties use protocol P_{k-1} to authenticate the message m_2, which is a computed hash value of m_1 using $i_{\mathcal{S}}^1$, and $i_{\mathcal{R}}^1$. Clearly, this protocol is perfectly complete.

Lemma 11. *Any computationally unbounded adversary has probability of at most ϵ to cheat the receiver into accepting a fraudulent message in protocol P_k.*

Proof. Given an execution of the protocol in which an adversary cheats the receiver into accepting a fraudulent message, it holds that $m_{\mathcal{S}}^1 \neq m_{\mathcal{R}}^1$ and $m_{\mathcal{S}}^k = m_{\mathcal{R}}^k$. Therefore, there exists an integer $1 \leq j \leq k-1$ such that $m_{\mathcal{S}}^j \neq m_{\mathcal{R}}^j$ and $m_{\mathcal{S}}^{j+1} = m_{\mathcal{R}}^{j+1}$. Denote this event by D_j. In what follows, we bound the probability of this event, showing $\Pr[D_j] \leq \frac{\epsilon}{2^{k-j}}$. Therefore, the adversary's cheating probability is at most $\sum_{j=1}^{k-1} \Pr[D_j] \leq \sum_{j=1}^{k-1} \frac{\epsilon}{2^{k-j}} < \epsilon$.

For any variable y in the protocol and for a given execution, let $T(y)$ be the time at which the variable y is fixed, i.e., $T(i_{\mathcal{R}}^j)$ denotes the time in which $\mathcal{R}$ sent the tag $i^j_{\mathcal{R}}$, and $T(\widehat{i}^j_{\mathcal{R}})$ denotes the time in which $\mathcal{S}$ received from the adversary the tag $\widehat{i}^j_{\mathcal{R}}$ corresponding to $i^j_{\mathcal{R}}$.

We assume here that j is odd, and refer the reader to the full version for the analysis in the case that j is even. There are three possible cases to consider:

1. $\boldsymbol{T(\widehat{i}^j_{\mathcal{R}}) < T(i^j_{\mathcal{R}})}$. In this case, the receiver chooses $i^j_{\mathcal{R}} \in_{\mathrm{R}} \mathrm{GF}[Q_j]$ only after the adversary chooses $\widehat{i}^j_{\mathcal{R}}$. Therefore,

$$\Pr\left[D_j\right] \leq \Pr_{i^j_{\mathcal{R}} \in_{\mathrm{R}} \mathrm{GF}[Q_j]}\left[i^j_{\mathcal{R}} = \widehat{i}^j_{\mathcal{R}}\right] = \frac{1}{Q_j} \leq \frac{\epsilon}{2^{k-j}} \enspace .$$

2. $\boldsymbol{T(\widehat{i}^j_{\mathcal{R}}) \geq T(i^j_{\mathcal{R}})}$ **and** $\boldsymbol{T(\widehat{i}^j_{\mathcal{S}}) \geq T(i^j_{\mathcal{S}})}$. In this case, the adversary chooses $\widehat{i}^j_{\mathcal{R}}$ not before the receiver chooses $i^j_{\mathcal{R}}$. If the adversary chooses $\widehat{i}^j_{\mathcal{R}} \neq i^j_{\mathcal{R}}$, then $m^{j+1}_{\mathcal{S}} \neq m^{j+1}_{\mathcal{R}}$, i.e., $\Pr\left[D_j\right] = 0$. Now suppose that the adversary chooses $\widehat{i}^j_{\mathcal{R}} = i^j_{\mathcal{R}}$. Since j is odd, the receiver chooses $i^j_{\mathcal{R}}$ only after he receives $\widehat{i}^j_{\mathcal{S}}$, therefore $T(i^j_{\mathcal{R}}) > T(\widehat{i}^j_{\mathcal{S}}) \geq T(i^j_{\mathcal{S}}) > T(m^j_{\mathcal{S}})$, and also $T(i^j_{\mathcal{R}}) > T(m^j_{\mathcal{R}})$. This means that $i^j_{\mathcal{R}}$ is chosen when $m^j_{\mathcal{R}}, \widehat{i}^j_{\mathcal{S}}, m^j_{\mathcal{S}}$ and $i^j_{\mathcal{S}}$ are fixed. Since $m^j_{\mathcal{S}} \neq m^j_{\mathcal{R}}$ and by the fact that for any choice of $i^j_{\mathcal{S}}$ and $\widehat{i}^j_{\mathcal{S}}$ the polynomials $C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{R}}) + \widehat{i}^j_{\mathcal{S}}$ and $C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{S}}) + i^j_{\mathcal{S}}$ are different as functions of $i^j_{\mathcal{R}}$, it follows that

$$\begin{aligned}\Pr\left[D_j\right] &\leq \Pr_{i^j_{\mathcal{R}} \in_{\mathrm{R}} \mathrm{GF}[Q_j]}\left[C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{S}}) + i^j_{\mathcal{S}} = C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{R}}) + \widehat{i}^j_{\mathcal{S}}\right] \\ &\leq \frac{1}{Q_j}\left\lceil \frac{n_j}{\log Q_j} \right\rceil \leq \frac{\epsilon}{2^{k-j}} \enspace .\end{aligned}$$

3. $\boldsymbol{T(\widehat{i}^j_{\mathcal{R}}) \geq T(i^j_{\mathcal{R}})}$ **and** $\boldsymbol{T(\widehat{i}^j_{\mathcal{S}}) < T(i^j_{\mathcal{S}})}$. As in the previous case, we can assume that the adversary chooses $\widehat{i}^j_{\mathcal{R}} = i^j_{\mathcal{R}}$. It always holds that $T(i^j_{\mathcal{S}}) > T(m^j_{\mathcal{S}})$ and $T(i^j_{\mathcal{R}}) > T(m^j_{\mathcal{R}})$. Since j is odd, the receiver sends $i^j_{\mathcal{R}}$ only after he receives $\widehat{i}^j_{\mathcal{S}}$, and therefore we can assume $T(\widehat{i}^j_{\mathcal{S}}) < T(i^j_{\mathcal{R}}) < T(i^j_{\mathcal{S}})$. This implies that the sender chooses $i^j_{\mathcal{S}} \in_{\mathrm{R}} \mathrm{GF}[Q_j]$ when $m^j_{\mathcal{S}}$, $\widehat{i}^j_{\mathcal{S}}$, $m^j_{\mathcal{R}}$ and $i^j_{\mathcal{R}}$ are fixed. Hence,

$$\Pr\left[D_j\right] \leq \Pr_{i^j_{\mathcal{S}} \in_{\mathrm{R}} \mathrm{GF}[Q_j]}\left[i^j_{\mathcal{S}} = C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{R}}) + \widehat{i}^j_{\mathcal{S}} - C^j_{i^j_{\mathcal{R}}}(m^j_{\mathcal{S}})\right] = \frac{1}{Q_j} \leq \frac{\epsilon}{2^{k-j}} \enspace .$$

$\square$

The following claims conclude this section by showing that our choice of parameters guarantees that in protocol P_k the sender manually authenticates at most $2\log(1/\epsilon) + 2\log^{(k-1)} n + \mathrm{O}(1)$ bits. We first show that the length n_{j+1} of the fingerprint computed in round j is roughly logarithmic in the length n_j of the fingerprint computed in round $j-1$, and then we use this fact to upper bound the length n_k of the manually authenticated fingerprint. The reader is referred to the full version of the paper for more details.

Claim 12. *If for every $1 \leq j \leq k-2$ it holds that $n_j > \frac{2^{k-j}}{\epsilon}$, then $n_{k-1} \leq \max\{4\log^{(k-2)} n_1 + 4\log 5 + 3,\ 27\}$.*

Claim 13. *The sender manually authenticates at most* $2\log(1/\epsilon)+2\log^{(k-1)} n+\mathrm{O}(1)$ *bits in protocol* P_k.

6 Lower Bound in the Manual Channel Model

In this section we prove a lower bound on the length of the manually authenticated string. We present here the proof for the simplified case of a perfectly complete 3-round protocol where $n \geq 3\log(1/\epsilon)$. The general proof is based on the same analysis, and is described in the full version of the paper. Moreover, note that by adding two more rounds, we can also assume for simplicity that in the last round the sender does not send an authentication tag x_i over the insecure channel (i.e., in the last round the sender *only* manually authenticates some string s). We prove the following theorem:

Theorem 14. *For any perfectly complete* $(n, \ell, 3, \epsilon)$*-authentication protocol in the manual channel model, where no authentication tag is sent in the last round, if* $n \geq 3\log(1/\epsilon)$*, then* $\ell \geq 2\log(1/\epsilon) - 2$.

As mentioned in Section 3, when the input message m is chosen uniformly at random, the honest execution of the protocol defines a probability distribution on the message m, the authentication tag x_1 (sent by the sender in the first round together with m), the authentication tag x_2 (sent by the receiver in the second round), and the manually authenticated string s (sent by the sender in the third round). We denote by M, X_1, X_2 and S the corresponding random variables.

The main idea of this proof is representing the entropy of the manually authenticated string S by splitting it as follows:

$$\begin{aligned}
\mathrm{H}(S) &= (\mathrm{H}(S) - \mathrm{H}(S|M, X_1)) + (\mathrm{H}(S|M, X_1) - \mathrm{H}(S|M, X_1, X_2)) \\
&\quad + \mathrm{H}(S|M, X_1, X_2) \\
&= \mathrm{I}(S; M, X_1) + \mathrm{I}(S; X_2|M, X_1) + \mathrm{H}(S|M, X_1, X_2) \ .
\end{aligned}$$

This representation captures the reduction of $\mathrm{H}(S)$ during the execution of the protocol, and allows us to prove that both the sender and the receiver must each independently reduce this entropy by at least $\log(1/\epsilon) - 1$ bits. We prove this by considering two possible man-in-the-middle attacks on the given protocol. In these attacks we use the fact that the adversary is computationally unbounded in that she can sample distributions induced by the protocol. For example, in the first attack, the adversary samples the distribution of X_2 given M, X_1 and S. While the distribution of X_2 given only M and X_1 can be sampled by merely following the protocol, this is not the case when sampling the distribution of X_2 given M, X_1 and S.

Lemma 15. *If* $n \geq 2\log\frac{1}{\epsilon}$*, then* $\mathrm{I}(S; M, X_1) + \mathrm{H}(S|M, X_1, X_2) > \log\frac{1}{\epsilon} - 1$.

Proof. Consider the following attack:

1. The adversary $\mathcal{A}$ chooses $\widehat{m} \in_{\mathrm{R}} \{0,1\}^n$ and runs an honest execution with the receiver. Denote by s the manually authenticated string fixed by this execution. Now, $\mathcal{A}$'s goal is to cause the sender to manually authenticate this string.
2. $\mathcal{A}$ chooses $m \in_{\mathrm{R}} \{0,1\}^n$ as the sender's input, and receives x_1 from the sender.
3. If $\Pr[m, x_1, s] = 0$ in an honest execution, then $\mathcal{A}$ quits (in this case $\mathcal{A}$ has zero probability in convincing the sender to manually authenticate s). Otherwise, $\mathcal{A}$ samples $\widehat{x}_2$ from the distribution of X_2 given (m, x_1, s), and sends it to the sender. The sender manually authenticates some string.
4. If the sender did not authenticate s, $\mathcal{A}$ quits. Otherwise, $\mathcal{A}$ forwards s to the receiver.

By the unforgeability requirement of the protocol we obtain:

$$\epsilon \geq \Pr[\mathcal{R} \text{ accepts and } m \neq \widehat{m}] \geq \Pr[\mathcal{R} \text{ accepts}] - 2^{-n} \ .$$

Therefore, the assumption $n \geq 2\log\frac{1}{\epsilon}$ implies that $\Pr[\mathcal{R} \text{ accepts}] < 2\epsilon$. Now we analyze the probability that the receiver accepts. Notice that:

- m and x_1 are chosen independently of s.
- $\widehat{x}_2$ is chosen conditioned on m, x_1 and s.
- The manually authenticated string sent by the sender is chosen conditioned on m, x_1 and x_2.

Therefore[5],

$$\begin{aligned}
\Pr[\mathcal{R} \text{ accepts}] &= \sum_{\substack{m, x_1, \widehat{x}_2, s: \\ \Pr[m, x_1, \widehat{x}_2, s] > 0}} \Pr[s] \Pr[m, x_1] \Pr[\widehat{x}_2 | m, x_1, s] \Pr[s | m, x_1, \widehat{x}_2] \\
&= \sum_{\substack{m, x_1, \widehat{x}_2, s: \\ \Pr[m, x_1, \widehat{x}_2, s] > 0}} \Pr[m, x_1, s] \frac{\Pr[s]}{\Pr[s | m, x_1]} \Pr[\widehat{x}_2 | m, x_1, s] \Pr[s | m, x_1, \widehat{x}_2] \quad (6.1) \\
&= \sum_{\substack{m, x_1, \widehat{x}_2, s: \\ \Pr[m, x_1, \widehat{x}_2, s] > 0}} \Pr[m, x_1, \widehat{x}_2, s] \, 2^{-\left\{\log \frac{\Pr[s | m, x_1]}{\Pr[s]} + \log \frac{1}{\Pr[s | m, x_1, \widehat{x}_2]}\right\}} \ ,
\end{aligned}$$

where Equation (6.1) follows from Bayes' rule. By Jensen's inequality,

$$\begin{aligned}
\Pr[\mathcal{R} \text{ accepts}] &\geq 2^{-\sum_{\substack{m, x_1, \widehat{x}_2, s: \\ \Pr[m, x_1, \widehat{x}_2, s] > 0}} \Pr[m, x_1, \widehat{x}_2, s] \left\{\log \frac{\Pr[s | m, x_1]}{\Pr[s]} + \log \frac{1}{\Pr[s | m, x_1, \widehat{x}_2]}\right\}} \\
&= 2^{-\{\mathrm{I}(S; M, X_1) + \mathrm{H}(S | M, X_1, X_2)\}} \ ,
\end{aligned}$$

and therefore $\mathrm{I}(S; M, X_1) + \mathrm{H}(S | M, X_1, X_2) > \log\frac{1}{\epsilon} - 1$. □

Lemma 16. *If* $n \geq 3\log\frac{1}{\epsilon}$ *and* $\ell < 2\log(1/\epsilon) - 2$, *then* $\mathrm{I}(S; X_2 | M, X_1) > \log\frac{1}{\epsilon} - 1$.

[5] For any random variable Z we write $\Pr[z]$ instead of $\Pr[Z = z]$.

Proof. Consider the following attack:

1. $\mathcal{A}$ chooses $m \in_{\mathrm{R}} \{0,1\}^n$, as the sender's input, and runs an honest execution with the sender. At the end of this execution, the sender manually authenticates a string s. $\mathcal{A}$ reads s, and delays it. Now, $\mathcal{A}$'s goal is to cause the receiver to accept this string together with a different input message $\widehat{m}$.
2. $\mathcal{A}$ samples $(\widehat{m}, \widehat{x}_1)$ from the joint distribution of (M, X_1) given s, and sends them to the receiver, who answers x_2.
3. If $\Pr[\widehat{m}, \widehat{x}_1, x_2, s] = 0$, then $\mathcal{A}$ quits. Otherwise, $\mathcal{A}$ forwards s to the receiver.

As in Lemma 15, $\epsilon \geq \Pr[\mathcal{R} \text{ accepts}] - \Pr[\widehat{m} = m]$. However, in this attack, unlike the previous attack, the messages m and $\widehat{m}$ are not chosen uniformly at random and independently. First m is chosen uniformly at random, then s is picked from the distribution of S given m, and then $\widehat{m}$ is chosen from the distribution of M given s. Therefore,

$$\begin{aligned}
\Pr[\widehat{m} = m] &= \sum_s \Pr[s] \sum_m (\Pr[m|s])^2 \leq \sum_s \Pr[s] \max_m \Pr[m|s] \sum_m \Pr[m|s] \\
&= \sum_s \Pr[s] \max_m \Pr[m|s] = \sum_s \max_m \Pr[m, s] \leq \sum_s \max_m \Pr[m] \quad .
\end{aligned}$$

Since the distribution of messages is uniform, and the authenticated string takes at most 2^ℓ values, we obtain $\Pr[\widehat{m} = m] \leq 2^{-n+\ell}$. From the assumptions that $\ell < 2\log(1/\epsilon) - 2$ and $n \geq 3\log(1/\epsilon)$ we get that $\Pr[\widehat{m} = m] < \epsilon$, and therefore $\Pr[\mathcal{R} \text{ accepts}] < 2\epsilon$. Now we analyze the probability that the receiver accepts. Notice that,

- $\widehat{m}$ and $\widehat{x}_1$ are chosen conditioned on s.
- x_2 is chosen conditioned only on $\widehat{m}$ and $\widehat{x}_1$.

Therefore,

$$\begin{aligned}
\Pr[\mathcal{R} \text{ accepts}] &= \sum_{\widehat{m}, \widehat{x}_1, s} \Pr[\widehat{m}, \widehat{x}_1, s] \sum_{\substack{x_2: \\ \Pr[\widehat{m}, \widehat{x}_1, x_2, s] > 0}} \Pr[x_2 | \widehat{m}, \widehat{x}_1] \\
&= \sum_{\substack{\widehat{m}, \widehat{x}_1, x_2, s: \\ \Pr[\widehat{m}, \widehat{x}_1, x_2, s] > 0}} \Pr[\widehat{m}, \widehat{x}_1, x_2, s] \frac{\Pr[x_2 | \widehat{m}, \widehat{x}_1]}{\Pr[x_2 | \widehat{m}, \widehat{x}_1, s]} \qquad (6.2) \\
&= \sum_{\substack{\widehat{m}, \widehat{x}_1, x_2, s: \\ \Pr[\widehat{m}, \widehat{x}_1, x_2, s] > 0}} \Pr[\widehat{m}, \widehat{x}_1, x_2, s]\, 2^{-\log \frac{\Pr[x_2 | \widehat{m}, \widehat{x}_1, s]}{\Pr[x_2 | \widehat{m}, \widehat{x}_1]}} \quad ,
\end{aligned}$$

where Equation (6.2) follows from Bayes' rule. By Jensen's inequality,

$$\Pr[\mathcal{R} \text{ accepts}] \geq 2^{-\sum_{\substack{\widehat{m}, \widehat{x}_1, x_2, s: \\ \Pr[\widehat{m}, \widehat{x}_1, x_2, s] > 0}} \Pr[\widehat{m}, \widehat{x}_1, x_2, s] \log \frac{\Pr[x_2 | \widehat{m}, \widehat{x}_1, s]}{\Pr[x_2 | \widehat{m}, \widehat{x}_1]}} = 2^{-\mathrm{I}(S; X_2 | M, X_1)} \quad ,$$

and therefore $\mathrm{I}(S; X_2 | M, X_1) > \log \frac{1}{\epsilon} - 1$. ☐

Now, Theorem 14 can be derived as follows. Suppose for contradiction that there exists a perfectly complete $(n, \ell, 3, \epsilon)$-authentication protocol, where no authentication tag is sent in the last round, and $n \geq 3\log(1/\epsilon)$ but $\ell < 2\log(1/\epsilon) - 2$. By using the fact that $\ell \geq \mathrm{H}(S)$, we can easily derive a contradiction: The above mentioned representation of $\mathrm{H}(S)$ and Lemmata 15 and 16 imply that $\mathrm{H}(S) > 2\log(1/\epsilon) - 2$. Therefore $\ell \geq 2\log(1/\epsilon) - 2$ in any such protocol. This concludes the proof of Theorem 14.

7 Lower Bound in the Shared Key Model

In this section we prove a lower bound on the entropy of the shared key. This lower bound settles an open question posed by Gemmell and Naor [9], and shows that the authentication protocol proposed by Gemmell and Naor is essentially optimal with respect to the entropy of the shared key.

We present here the result for the simplified case of a perfectly complete 3-round protocol. The general proof is based on the same analysis, and is described in the full version. We prove the following theorem:

Theorem 17. *For any perfectly complete $(n, \ell, 3, \epsilon)$-authentication protocol in the shared key model, it holds that $\mathrm{H}(S) \geq 2\log(1/\epsilon)$, where S is the ℓ-bit shared key.*

As mentioned in Section 3, when the shared key s is chosen from its specified distribution, and the input message m is chosen uniformly at random, the honest execution of the protocol defines a probability distribution on the shared key s, the message m, the authentication tag x_1 (sent by the sender in the first round together with m), the authentication tag x_2 (sent by the receiver in the second round), and the authentication tag x_3 (sent by the sender in the third round). We denote by S, M, X_1, X_2 and X_3 the corresponding random variables.

We apply again the proof technique described in Section 6, and represent the entropy of the shared key S by splitting it as follows (we refer the reader to the full version for more details):

$$\begin{aligned}\mathrm{H}(S) &= \mathrm{I}(S; M, X_1) + \mathrm{I}(S; X_2|M, X_1) + \mathrm{I}(S; X_3|M, X_1, X_2) \\ &\quad + \mathrm{H}(S|M, X_1, X_2, X_3) \ .\end{aligned}$$

Lemma 18. $\mathrm{I}(S; M, X_1) + \mathrm{I}(S; X_3|M, X_1, X_2) \geq \log \frac{1}{\epsilon}$.

Lemma 19. $\mathrm{I}(S; X_2|M, X_1) + \mathrm{H}(S|M, X_1, X_2, X_3) \geq \log \frac{1}{\epsilon}$.

8 Breaking the Lower Bounds Implies One-Way Functions

In this section we prove Theorem 10, namely, we show that in the computational setting one-way functions are essential for the existence of protocols breaking the lower bound stated in Theorem 8. As in Section 6 we prove here the result only for 3-round protocols, where in the last round no authentication tag x_i is sent over the insecure channel. Moreover, for simplicity we also assume that $n \geq 1/\epsilon$, and refer the reader to the full version for the proof of the general statement.

Theorem 20. *In the manual channel model, if there exists a computationally secure perfectly complete $(n, \ell, k, \epsilon, t)$-authentication protocol where no authentication tag is sent in the last round, such that $n \geq 1/\epsilon$, $\ell < 2\log(1/\epsilon) - 4$ and $t = \Omega(poly(n, k))$, then one-way functions exist.*

Proof. We show that if one-way functions do not exist, then the attacks described in Section 6 can be carried out by a polynomial-time adversary with almost the same success probability. We first focus on the attack described in Lemma 15.

Let f be a function defined as follows: f takes as input three strings $r_{\mathcal{S}}$, $r_{\mathcal{R}}$ and m, and outputs (m, x_1, x_2, s) – the transcript of the protocol, where $r_{\mathcal{S}}$, $r_{\mathcal{R}}$ and m are the random coins of the sender, the random coins of the receiver, and the input message, respectively. Let f' denote the function that is obtained from f by eliminating its third output, i.e., $f'(r_{\mathcal{S}}, r_{\mathcal{R}}, m) = (m, x_1, s)$. If one-way functions do not exist, then also distributionally one-way functions do not exist. Therefore, for any constant $c > 0$ there exists a probabilistic polynomial-time Turing machine $\mathcal{M}$ that on input (m, x_1, s) produces a distribution that is n^{-c}-statistically close to the uniform distribution on all the pre-images of (m, x_1, s) under f'. The polynomial-time adversary will use this $\mathcal{M}$ in the attack.

Let $\mathcal{A}$ denote the unbounded adversary that carried the attack described in Lemma 15, and let $\mathcal{A}^{\mathrm{PPT}}$ denote a polynomial-time adversary that carries the following attack (our goal is that the receiver will not be able to distinguish between $\mathcal{A}$ and $\mathcal{A}^{\mathrm{PPT}}$):

1. $\mathcal{A}^{\mathrm{PPT}}$ chooses $\widehat{m} \in_{\mathrm{R}} \{0,1\}^n$ and runs an honest execution with the receiver. Denote by s the manually authenticated string fixed by this execution.
2. $\mathcal{A}^{\mathrm{PPT}}$ chooses $m \in_{\mathrm{R}} \{0,1\}^n$ as the sender's input, and receives x_1 from the sender.
3. $\mathcal{A}^{\mathrm{PPT}}$ executes $\mathcal{M}$ on input (m, x_1, s), and then applies f to $\mathcal{M}$'s answer to compute $\widehat{x}_2$ and send it to the sender. The sender manually authenticates some string s^*.
4. $\mathcal{A}^{\mathrm{PPT}}$ forwards s^* to the receiver (who must accept $\widehat{m}$ if $s^* = s$ by the perfect completeness).

Let $\mathrm{Prob}^{\mathcal{R}}$ and $\mathrm{Prob}^{\mathrm{PPT},\mathcal{R}}$ denote the probabilities that the receiver $\mathcal{R}$ accepts $\widehat{m}$ when interacting with $\mathcal{A}$ and when interacting with $\mathcal{A}^{\mathrm{PPT}}$, respectively. Then, from the point of view of the receiver, the only difference in the these two executions is in the distribution of s^*. Therefore, $|\mathrm{Prob}^{\mathcal{R}} - \mathrm{Prob}^{\mathrm{PPT},\mathcal{R}}|$ is at most twice the statistical distance between s^* in the interaction with $\mathcal{A}$ and s^* in the interaction with $\mathcal{A}^{\mathrm{PPT}}$. By the above mentioned property of $\mathcal{M}$, this statistical difference is at most n^{-c}. Therefore, for sufficiently large t, we obtain as in Lemma 15

$$2\epsilon > \mathrm{Prob}^{\mathrm{PPT},\mathcal{R}} \geq \mathrm{Prob}^{\mathcal{R}} - 2n^{-c} \geq 2^{-\{\mathrm{I}(S;M,X_1)+\mathrm{H}(S|M,X_1,X_2)\}} - 2n^{-c} \ .$$

In particular, and since $n \geq 1/\epsilon$, we can choose the constant c such that $2n^{-c} < \epsilon$, and obtain $\mathrm{I}(S; M, X_1) + \mathrm{H}(S|M, X_1, X_2) > \log \frac{1}{\epsilon} - 2$.

A similar argument applied to the attack described in Lemma 16 yields $\mathrm{I}(S; X_2|M, X_1) > \log \frac{1}{\epsilon} - 2$, and therefore $\mathrm{H}(S) > 2 \log \frac{1}{\epsilon} - 4$. $\square$

Acknowledgments. We thank Benny Pinkas, Danny Segev, Serge Vaudenay and the anonymous referees for their remarks and suggestions.

References

1. B. Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd FOCS*, pages 345–355, 2002.
2. Bluetooth. http://www.bluetooth.com/bluetooth/.
3. Certified Wireless USB. http://www.usb.org/developers/wusb/.
4. G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. In *30th STOC*, pages 141–150, 1998.
5. G. Di Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. Efficient and non-interactive non-malleable commitment. In *EUROCRYPT '01*, pages 40–59, 2001.
6. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.
7. C. Gehrmann. Cryptanalysis of the Gemmell and Naor multiround authentication protocol. In *CRYPTO '94*, pages 121–128, 1994.
8. C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA Cryptobytes*, 7:29–37, 2004.
9. P. Gemmell and M. Naor. Codes for interactive authentication. In *CRYPTO '93*, pages 355–367, 1993.
10. E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *Bell System Technical Journal*, 53(3):405–424, 1974.
11. R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *30th FOCS*, pages 230–235, 1989.
12. S. Laur, N. Asokan, and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. Cryptology ePrint Archive, Report 2005/424, 2005.
13. U. M. Maurer. Authentication theory and hypothesis testing. *IEEE Transactions on Information Theory*, 46(4):1350–1356, 2000.
14. M. Naor and G. N. Rothblum. The complexity of online memory checking. In *46th FOCS*, pages 573–584, 2005.
15. M. Naor, G. Segev, and A. Smith. Tight bounds for unconditional authentication protocols in the manual channel and shared key models. Cryptology ePrint Archive, Report 2006/175, 2006.
16. R. Pass and A. Rosen. New and improved constructions of non-malleable cryptographic protocols. In *37th STOC*, pages 533–542, 2005.
17. G. J. Simmons. Authentication theory/coding theory. In *CRYPTO '84*, pages 411–431, 1984.
18. G. J. Simmons. The practice of authentication. In *EUROCRYPT '85*, pages 261–272, 1985.
19. S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *CRYPTO '05*, pages 309–326, 2005.
20. M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981.