Efficient Mutual Data Authentication Using Manually Authenticated Strings

Sven Laur[1], N. Asokan[2], and Kaisa Nyberg[1,2]

[1] Helsinki University of Technology, Finland
[2] Nokia Research Center, Finland
{sven.laur,kaisa.nyberg}@tkk.fi,
{n.asokan,kaisa.nyberg}@nokia.com

Abstract. Solutions for an easy and secure setup of a wireless connection between two devices are urgently needed for WLAN, Wireless USB, Bluetooth and similar standards for short range wireless communication. In this paper we analyse the SAS protocol by Vaudenay and propose a new three round protocol MA-3 for mutual data authentication based on a cryptographic commitment scheme and short manually authenticated out-of-band messages. We show that non-malleability of the commitment scheme is essential for the security of the SAS and the MA-3 schemes and that extractability or equivocability do not imply non-malleability. We also give new proofs of security for the SAS and MA-3 protocols and suggestions how to instantiate the MA-3 protocol in practise.

1 Introduction

The pairing problem. In quest for easy and secure setup of a wireless connection between two devices many new solutions have recently been, or are currently being developed. Most urgently such a solution is needed for WLAN, as homes are being equipped with WLAN access points; the home users should have a clear and manageable procedure to set up a secure WLAN which makes it easy to add and remove devices from the network. The Wireless Protected Access (WPA) by Wi-Fi Alliance provides encryption and authentication but is still today delivered with a common secret key for all users. The WiFi Alliance is working on a more secure solution [KW04,INQ05]. Also the Wireless USB has launched a similar initiative on, what is called as, Association Models [Hun05]. For Bluetooth the problem of secure connection set up has a longer history [GPS04]. The standard Bluetooth Pairing mechanism is based on symmetric cryptography, and is typically not very well implemented. On the other hand, experiences from Bluetooth have shown what kind of the security and usability requirements one is faced with.

Using Diffie-Hellman or some other public key based key exchange the problem of establishing a shared secret over an insecure wireless channel is reduced to the problem of preventing an active online man-in-the-middle, that is, to the problem of authentication of the public keys. It is common to assume existence of some auxiliary user operated communication channel, over which some limited amount of confidential or authenticated information is exchanged between the devices. For the evolution of manual data authentication and authenticated key exchange protocols see

[GMN04,Hoe05,Vau05]. The auxiliary channels, also called as out-of-band (OOB) channels, can be classified according to the types of interfaces they use with the devices. Basically, as interfaces are either input or output, there are three possible combinations for the device interfaces used at the ends of the OOB channel: input-input (I/I), output-input (O/I), output-output (O/O). Given such channels, possibly with different capabilities, bandwidths and security, one can develop suitable operations to authenticate data for the devices, or what is the same, verify that some piece of data established over the insecure channel is the same in both devices. In [GMN04] three authentication protocols, one for each basic interface combination was presented: MANA I using O/I device interfaces, MANA II for O/O device interfaces, and MANA III for I/I device interfaces. MANA I and MANA II use short message authentication codes computed form the data and compared over the channel. In MANA III a shared secret password is entered to both devices and randomised verification takes place over the insecure channel. The probability of success of a man-in-the-middle in these protocols is about $2^{-\ell}$, if ℓ is the length of the message authentication code in MANA I and MANA II, or the length of the password in MANA III. However, MANA I and MANA II are not optimal, in the sense that the string to be compared must also include the key, also of length ℓ, for the message authentiction code. It is also important to note that MANA I and MANA III require that the OOB channel preserves confidentiality, while MANA II requires only an authenticated OOB channel.

Protocols that are suitable to handle use cases with I/I interfaces can be easily transformed to protocols for O/I scenario. Instead of entering a password to the devices, it can be generated in one device which outputs it to be entered to the second device. Similarly, an O/O scenario can be adapted to O/I interfaces be giving the task of comparing the values to the device with input-interface. Therefore, a set of two protocols, where one protocol is suited for I/I scenario with a secret password, and another one handles the O/O scenario with a short authenticated string, covers the three basic interface scenarios.

For I/I scenario, there is a wealth of protocols known as password based key agreement protocols, e.g. [BM92,KOY01]. Being designed for the client-server authentication these protocols are designed to allow reuse of the password. This is not a necessary requirement when setting up a secure connection between two peer devices, and typically increases computational complexity of the procedure. Complexity is further incresed by implementing secure storage of secrets by the client and the server. The basic EKE protocol [BM92] without password protection does not have unnecessary complexity and is well suited for secure connection set up between peer devices. MANA III provides another good solution to the problem. The scenario using output interfaces in both devices has some advantages, as the user need not enter any random strings, but only compare them. Therefore it is forseen that a manual authentication protocol will be required to support this scenario. Recently, Vaudenay presented a manual authentication protocol using short authenticted strings [Vau05]. From the point-of-view of interfaces, as well as security, Vaudenay's SAS protocol is similar to MANA II. However, it provides significant improvement over MANA II in two aspects. First, the length of the string to be verified in authenticated manner is optimal, that is, one half of the length of the string used by MANA II for the same security level. The second improvement is that the operations performed by the user is reduced. In MANA II the devices must have the data ready in both devices before the verification can start, and the users must indicate the start in both devices. In this manner a "strong authenticated channel" as it is called in [Vau05] is established between the devices. In the SAS protocol this step is not visible to the user thanks to a cryptographic commitment scheme.

Our contributions. The SAS protocol for unilateral authentication has three moves over the insecure channel, and the combined protocol for manual cross-authentication of data takes four moves [Vau05], Annex A. In this paper we show that the number of moves can be reduced to three. This is interesting in theory, but is important also in practise as key agreement by two peer devices typically requires mutual authentication. We also show that the SAS and our new message authentication protocol MA-3 depend heavily on non-malleability of a commitment scheme: one has to give an explicit white-box security proof unless he explicitly assumes non-malleability of the commitment scheme. As the MA-3 protocol is a general representation of three round protocol where messages are independent from the preceding reply, any other construction without non-malleability requirement must be more involved.

Road-map. Section 2 contains rather lengthy but necessary characterisation of various flavours of commitment schemes and other cryptographic primitives. Section 3 contains description of the SAS and MA-3 protocols along with adversarial models. Section 4 contains constructive counter-examples to show that non-malleability is essential for security proofs. Section 5 contains security proofs for both protocols. Finally, Section 6 contains discussion what are reasonable choices for necessary cryptographic primitives.

2 Cryptographic preliminaries

Throughout the article we consider algorithms with a bounded working-time t, where t is at least proportional to the length of the program code. Notation $g(t) = \mathcal{O}(f(t))$ denotes asymptotic complexity w.r.t. t, i.e., $\limsup_{t\to\infty} g(t)/f(t) < \infty$. We denote independent random draws from a set $\mathcal{X}$ by $x \leftarrow \mathcal{X}$. Outputs of a randomised algorithm A are denoted by $x \leftarrow A$. Events are denoted by mnemonic names like events.

Keyed hash functions. Cryptographic hash functions are often used to assure data integrity. Shortly put, a keyed hash function $h : \mathcal{D} \times \mathcal{K} \to \mathcal{T}$ is a two-argument function where the first argument corresponds to a message and the second to a key. Keyed hash function h is ε-almost universal (denoted by ε-AU$_2$in terms of [Sti92]), if for any $x_0, x_1 \in \mathcal{D},\ x_0 \neq x_1$ the collision probability $\Pr\left[k \leftarrow \mathcal{K} : h(x_0, k) = h(x_1, k)\right] \leq \varepsilon$. A hash function h is uniform if for each x and y the probability that for a randomly chosen k we get $h(x, k) = y$ is $1/|\mathcal{T}|$.

Pseudorandom combiners. Combiner functions are used to combine different inputs into a single output. In the context of key-agreement protocols, combiners must assure randomness of the output even only a single input is chosen uniformly. A combiner $f : \mathcal{K}_1 \times \mathcal{K}_2 \to \mathcal{K}$ provides such goal if $f(r_1, \cdot)$ and $f(\cdot, r_2)$ are uniform functions for all $r_1 \in \mathcal{K}_1$ and $r_2 \in \mathcal{K}_2$. We call these functions left-right uniform combiners. A combiner is a (t, ε)-pseudorandom permutation if $f_k(x) = f(k, x)$ is a (t, ε)-pseudorandom permutation where the first argument plays a role of a secret key.

Commitment schemes. Formally, a commitment scheme is a triple of functionalities $\mathcal{C}om = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$. A setup algorithm Gen generates public parameters pk of the commitment scheme. The commitment function $\mathsf{Com}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \to \mathcal{C} \times \mathcal{D}$ transforms message $m \in \mathcal{M}$ into a short digest c and a decommitment value d. Usually $d = (m, s)$ where $s \in \mathcal{R}$ is the used randomness. Finally, correctly formed commitments can be opened, i.e., $\mathsf{Open}_{\mathsf{pk}}(c, d) = m$ for all $(c, d) = \mathsf{Com}_{\mathsf{pk}}(m, s)$. Incorrect decommitment values yield to a special abort value $\perp$. Binding and hiding properties are basic requirements for commitment schemes. A commitment scheme is (t, ε_1)-hiding [1] iff any t-time adversary A achieves advantage

$$\mathrm{Adv}^{\mathsf{hid}}(A) = 2 \cdot \left| \Pr\left[\begin{array}{l} \mathsf{pk} \leftarrow \mathcal{K}, b \leftarrow \{0,1\}, x_0 \leftarrow A(\mathsf{pk}), x_1 \leftarrow \mathcal{M}, \\ (c_i, d_i) = \mathsf{Com}_{\mathsf{pk}}(x_i, s_i), s_i \leftarrow \mathcal{R} : \ A(c_b) = b \end{array} \right] - \frac{1}{2} \right| \leq \varepsilon_1 \ .$$

A commitment scheme is (t, ε_2)-binding iff any t-time adversary A achieves advantage

$$\mathrm{Adv}^{\mathsf{bind}}(A) = \Pr\left[\begin{array}{l} \mathsf{pk} \leftarrow \mathcal{K}, (c, d_0, d_1) \leftarrow A(\mathsf{pk}) : \\ \perp \neq \mathsf{Open}_{\mathsf{pk}}(c, d_0) \neq \mathsf{Open}_{\mathsf{pk}}(c, d_1) \neq \perp \end{array} \right] \leq \varepsilon_2 \ .$$

Extractable commitment schemes. Extractable commitment schemes as first defined in [SCP00,Cre02] have slightly different setup algorithm Gen that returns a secret and a public key pair $(\mathsf{sk}, \mathsf{pk})$. Commitment and opening functionalities use only the public key and are defined as before. We say that the commitment scheme is (t, ε)-extractable if there is an efficient function $\mathsf{Extr}_{\mathsf{sk}} : \mathcal{C} \to \mathcal{M}$ that allows to extract messages from valid commitments, more specifically, for any t time adversary A

$$\mathrm{Adv}^{\mathsf{extr}}(A) = \Pr\left[\begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}, (c, d) \leftarrow A(\mathsf{pk}) : \\ \mathsf{Extr}_{\mathsf{sk}}(c) \neq \mathsf{Open}_{\mathsf{pk}}(c, d) \neq \perp \end{array} \right] \leq \varepsilon \ .$$

Obviously, the commitment scheme is not hiding any more if the secret key sk has been leaked out, as one can use sk to extract commitments.

Equivocable commitment schemes. An equivocable commitment scheme as defined in [CIO98,DG03] is a tuple of functions $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Com}^*, \mathsf{Equiv}, \mathsf{Open})$. A setup algorithm Gen returns a public and secret key pair $(\mathsf{pk}, \mathsf{sk})$. Commitment and opening functionalities use only the public key and are defined as before. Function $\mathsf{Com}^*_{\mathsf{sk}}$ provides a fake commitment c with the auxiliary information $\sigma \in \mathcal{S}$ such that c can be opened to any value using $\mathsf{Equiv}_{\mathsf{sk}} : \mathcal{M} \times \mathcal{C} \times \mathcal{S} \to \mathcal{D}$, i.e., for all $(c, \sigma) \leftarrow \mathsf{Com}^*_{\mathsf{sk}}$, $x \in \mathcal{M}$ we have $\mathsf{Open}_{\mathsf{pk}}(c, \mathsf{Equiv}_{\mathsf{sk}}(x, c, \sigma)) = x$. Secondly, it should be infeasible to distinguish between true and faked commitments. A commitment scheme is (t, ε)-equivocable iff for any t time adversary A

$$\mathrm{Adv}^{\mathsf{equiv}}(A) = |\Pr[A(\mathsf{pk}) = 1 | \mathsf{World}_1] - \Pr[A(\mathsf{pk}) = 1 | \mathsf{World}_0]| \leq \varepsilon$$

where World_0 and World_1 denote different environments. In both environments, Gen is run and pk is fed to A. Additionally, A can query commitments for $x \in \mathcal{M}$ in a

[1] Traditionally semantical security is defined via a left-or-right game but here it is convenient to use a real-or-random game. The two definitions are equivalent up to a small constant 2.

black-box way. In World_0, the corresponding reply is $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x, s)$, whereas in World_1, the pair (c, d) is computed as $(c, \sigma) \leftarrow \mathsf{Com}^*_{\mathsf{sk}}$ and $d \leftarrow \mathsf{Equiv}_{\mathsf{sk}}(x, c, \sigma)$. Again, the commitment scheme is not binding any more when sk has been leaked out.

Non-malleable commitment schemes. Many notions of non-malleable commitments have been proposed in cryptographic literature [CIO98,FF00,DG03] starting from the seminal article [DDN91] by Dolev, Dwork and Naor. All these definitions try to capture requirements that are necessary to defeat man-in-the-middle attacks. We adopt the modernised version of [CIO98]—non-malleability w.r.t. opening—that is slightly weaker than the definition [DG03]. Shortly put, we assume that committed messages are independent from pk that is clearly satisfied in the scope of the article. The choice allows to define non-malleability without a simulator using comparison based security similarly to the framework of non-malleable encryption [BS99]. First, note that the equivalence result between simulation and comparison based definition [BS99] directly generalises to commitments. [2] Secondly, the definition of non-malleable encryption is more strict (non-malleability w.r.t. commitment) and thus CCA secure encryption schemes can be used as non-malleable commitments provided that the public key pk is generated by the trusted party, i.e., non-malleability is achievable in the common reference string model. The latter is a relatively mild assumption in practice, as manufactures of electronic equipment can hardwire a common public key into all devices. Formally, an adversary is a quadruple $A = (A_1, A_2, A_3, A_4)$ of efficient algorithms where (A_1, A_2, A_3) represents an active part of the adversary that creates and afterwards tries to open related commitments and A_4 represents a target relation or a distinguisher. The relation A_4 is completely specified before seeing a decommitment of the challenge commitment. A succeeds if A_4 can distinguish between two environments World_0 and World_1. In both environments, Gen is run to produce pk and then

1. $A_1(\mathsf{pk})$ outputs a description of an efficient message sampler $\mathcal{M}_0$ and a state σ_1, then $x_0, x_1 \leftarrow \mathcal{M}_0$ are independently drawn.
2. Next, $A_2(c, \sigma_1)$ is run with $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_0, s),\ s \leftarrow \mathcal{R}$. A_2 outputs a state σ_2 and a commitment vector $(c_1, \ldots, c_n)$ with arbitrary length that does not contain c.
3. Now, $A_3(d, \sigma_2)$ must outputs a valid decommitment vector $(d_1, \ldots, d_n)$, i.e., all $y_i = \mathsf{Open}_{\mathsf{pk}}(c_i, d_i) \neq \perp$. If some $y_i = \perp$ then A is halted with 0. [3]
4. Finally, in the environment World_0 we invoke $A_4(x_0, y_1, \ldots, y_n, \sigma_2)$ whereas in World_1 we invoke $A_4(x_1, y_1, \ldots, y_n, \sigma_2)$.

A commitment scheme is (t, ε)-non-malleable iff for any t time adversary [4] A

$$\mathrm{Adv}^{\mathsf{nm}}(A) = |\Pr[A_4 = 1|\mathsf{World}_0] - \Pr[A_4 = 1|\mathsf{World}_1]| \leq \varepsilon \ .$$

CCA security of commitment schemes. A commitment scheme is secure under a chosen commitment attack if it is (t, ε_1)-hiding and (t, ε_2)-binding even if adversary A can

[2] Substitutions in the definitions and proofs of [BS99] are straightforward, except there is no decommitment oracle and an isolated sub-adversary A_3 has to compute decommitment values.

[3] The latter restriction is necessary, as otherwise A_3 can send n bits of information to A_4 by refusing to open some commitments. The same problem has been addressed [CKOS01,DG03] by requiring that behaviour of A_4 should not change if y_i is replaced with $\perp$. The latter is somewhat cumbersome as static program analysis is undecidable in theory.

[4] The sampling time of $x_0, x_1 \leftarrow \mathcal{M}_0$ is included in the working time of A.

use any decommitment oracle $\mathcal{O}_{\mathsf{dec}}$ in a nontrivial manner. Given a commitment c, oracle $\mathcal{O}_{\mathsf{dec}}(c)$ must return d such that $\mathsf{Open}_{\mathsf{pk}}(c, d) \neq \perp$ for all valid commitments c. Querying of the challenge commitment is not allowed. For perfectly binding commitments there is a unique opening, otherwise many possibilities emerge. Then it is conceptually easier to imagine that the adversary prescribes the target message x to $\mathcal{O}_{\mathsf{dec}}$ when scheme is non-binding. The latter makes the environment similar to equivocable commitments. However, for CCA secure commitment schemes, a single double decommitment should not jeopardise security of other commitments. Many non-binding commitment schemes like [Ped91,FO97] do not satisfy this requirement, i.e., CCA security of non-binding commitments is much more restrictive than equivocability.

Relations between commitment schemes. Obviously, any (t, ε)-non-malleable commitment scheme is $(\tau, 2\varepsilon)$-hiding and (τ, ε)-binding with $\tau = t - \mathcal{O}(1)$, since either A_2 or A_3 can send information about x_0 to A_4. Equivocability does not imply non-malleability as shown in Theorem 2. Although latter seems paradoxical, since historically non-malleable commitments were constructed from equivocable ones [CIO98], there is no contradiction, as authors use equivocable commitments in a more complex construction to achieve non-malleability. Extractability does not imply non-malleability and *vice versa*. Finally, CCA security implies non-malleability as shown in Theorem 8.

3 Manual authentication with short authenticated strings

3.1 Protocol description

In our setting, two honest parties Alice and Bob have non-confidential inputs m_a and m_b and they want to establish a shared common public output $m_a||m_b$ in a malicious environment. Besides in-band messages parties can send short out-of-band (OOB) messages in authenticated manner. The total length of OOB-messages should be as small as possible. On the other hand, short OOB-messages cannot provide negligible failure probability against man-in-the-middle attacks. In the following, we analyse and generalise the unilateral three round [5] message authentication protocol SAS by Vaudenay [Vau05] depicted on Fig. 1. Compared to the MANA II protocol [GMN04] depicted on Fig. 2 users do not have to confirm that data has arrived before the first message. On the other hand, security requirements for the used cryptographic primitives are more demanding.

1. Alice computes $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_a||r_a, s),\ s \leftarrow \mathcal{R}$ for $r_a \leftarrow \mathcal{K}$ and sends c to Bob.
2. Bob sends $r_b \leftarrow \mathcal{K}$ to Alice.
3. Alice sends the decommitment d to Bob and both compute $\mathtt{check} = r_a \oplus r_b$.
4. Bob accepts m_a iff the local control values $\mathtt{check}_a$ and $\mathtt{check}_b$ coincide.

Fig. 1. The SAS protocol

[5] Note that steps 3 and 4 can be combined into a single round.

1. Alice and Bob verify over OOB channel that the data m is has arrived to both parties.
2. Alice sends $k \leftarrow \mathcal{K}$ to Bob and both compute $\texttt{check} = h(m, k)||k$.
3. Both parties accept m iff the local control values $\texttt{check}_a$ and $\texttt{check}_b$ coincide.

Fig. 2. The MANA II protocol

The SAS protocol has a few minor shortcomings. First, the commitment incorporates non-confidential message m_a though the latter can be shortened to $h(m_a)$ where h is a collision resistant. Still the dependence between m_a and c is undesirable, as using longer commitments is more time-consuming. Secondly, the protocol does not provide mutual authentication. Running two copies of the SAS protocol yields a four round mutual authentication protocol but the latter is not round optimal. Also, the formal security proofs given by Vaudenay are slightly incorrect, see discussion in Subsection 4.2. These shortcomings inspired us to design and analyse a three round mutual authentication protocol depicted as Fig. 3 where h is a keyed hash function (MAC) and f pseudorandom permutation e.g. 128-bit AES encryption. The protocol is more modular: commitments are independent of messages, authenticity is guaranteed by the MAC similarly to MANA II. The effective bit size $\ell = \log_2 |\mathcal{T}|$ of control values $\texttt{check}$ determines achievable security. Since m_b might be computed after the first round, the MA-3 protocol can be naturally combined with any key-exchange protocol. See the discussion below about security in arbirary context.

1. Alice computes $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(r_a, s)$ for $s \leftarrow \mathcal{R}, r_a \leftarrow \mathcal{K}_a$ and sends (m_a, c) to Bob.
2. Bob sends $r_b \leftarrow \mathcal{K}_b$ and m_b to Alice.
3. Alice sends d to Bob and both compute $\texttt{check} = h(m_a||m_b, k)$ where $k = f(r_a, r_b)$.
4. Both parties accept $m = m_a||m_b$ iff the local test values $\texttt{check}_a$ and $\texttt{check}_b$ coincide.

Fig. 3. Three round mutual authentication protocol MA-3.

3.2 Adversarial models

Stand-alone model. We first analyse the stand-alone model where no other protocols are executed. Let Charley be a malicious courier that transfers messages form Alice to Bob and vice versa. Let c, m_a, m_b, r_b, d denote messages received by Charlie and c', m'_a, m'_b, r'_b, d' potentially altered messages received by Alice and Bob. Let $\texttt{check}_a$ and $\texttt{check}_b$ denote final control values obtained by Alice and Bob. Charlie succeeds in deception if $\texttt{check}_a = \texttt{check}_b$ although $m_a||m'_b \neq m'_a||m_b$. As the control values $\texttt{check}$ are ℓ-bit short, the probability of random guessing is $2^{-\ell}$. Therefore, we cannot guarantee deception probability below $2^{-\ell}$. Our aim is to prove that deception probability is negligibly bigger than $2^{-\ell}$.
Security in arbitrary context. Classical composition theorems [Gol04] assure that one can sequentially compose protocols that are secure in stand-alone model and the result-

ing protocol has only a cumulative security drop. Though the latter is *generally* not true for concurrent composition, we can prove that both the SAS and the MA-3 protocols are secure in any computational context if Alice and Bob follow the protocol and values r_a and r_b are not used in other protocols. Due to the lack of space we do not formalise the claim completely. Essentially, if Alice and Bob follow the protocol and do not use $m_a||m_b$ in the computations before the end of the authentication, then protocol messages can be perfectly simulated by the adversary himself, as m_a and m_b are public. Compared to the ideal implementation, where adversary can only observe messages and decide whether to drop them or not, we loose ε in security for each message. Batch authentication of several messages can be used to preserve security level. In particular, all key-exchange protocols can be secured by authenticating a protocol transcript $(m_1, m_2, \dots, m_k)$. Since the transcript must be fixed before the third round, the secured protocol has one extra round and we loose only ε in security, except for single round protocols that have two extra rounds.

4 Necessary requirements to building-blocks

Next, we derive minimal security requirements for building-blocks that are necessary to prove security of the MA-3 and SAS protocols using standard black-box reductions.

4.1 Mutual authentication

Note that any three round mutual authentication protocol where second and third messages are independent from previous replies has a form depicted in Fig. 3, since knowledge of first and third message must allow to compute r_a and from the second message it must be possible to compute r_b. Therefore, following analysis provides quite general characterisation of properties required from $\mathcal{C}om$, f and h. In the following, we assume that f is a left-right uniform combiner instead of pseudorandom permutation. The assumption is not essential, rather a natural simplification. Since the set of possible control values $\mathcal{T}$ is small, the hash function h must satisfy unconditional security guarantees. As the success of simple substitution attack is $\Pr\left[h(m_a||m'_b, k) = h(m'_a||m_b, k)\right]$, then h must be at least ε-almost universal. A well known lower bound [Sar80] on ε states that $\varepsilon \geq \frac{|\mathcal{D}|-|\mathcal{T}|}{|\mathcal{T}|(|\mathcal{D}|-1)}$ for any hash function family $h : \mathcal{D} \times \mathcal{K} \to \mathcal{T}$ provided that keys are chosen randomly and thus the lower bound on failure is indeed $\varepsilon \gtrsim 2^{-\ell}$ for all practical message sizes. Next, we show that a specific form non-malleability of the commitment scheme is necessary. We construct a specific hash function such that flipping a last bit allows successful deception and then we convert an ordinary commitment scheme into a malleable one that permits the necessary bit flip. Let h_0 be a hash function with key space $\mathcal{K}_0$. Given two different target messages $m_0, m_1 \in \mathcal{D}$, define h with extended key space $\mathcal{K} = \mathcal{K}_0 \times \{0, 1\}$ by the following rule

$$h(m, k||b) = \begin{cases} h_0(m, k), & \text{if } m \notin \{m_0, m_1\} \\ h_0(m_{i \oplus b}, k), & \text{otherwise} \end{cases} ,\quad .$$

Theorem 1. *If h_0 is a ε-almost universal then h is a ε-almost universal.* □

Let $\mathcal{C}om = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$ be a commitment scheme with message space $\mathcal{K}$ and let $g : \mathcal{K} \times \mathcal{C} \to \mathcal{K}$ be an efficient deterministic function. Fix $\mathcal{C}^\circ = \{0,1\} \times \mathcal{C}$ and define $\mathsf{Com}^\circ_{\mathsf{pk}} : \mathcal{K} \times \mathcal{R} \to \mathcal{C}^\circ \times \mathcal{D}$ and $\mathsf{Open}^\circ_{\mathsf{pk}}$ procedures in the following way

$$(c_\circ, d) \leftarrow \mathsf{Com}^\circ_{\mathsf{pk}}(x, s), \quad c_\circ = (b, c), \quad b = 0, \quad (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x, s),$$

$$\mathsf{Open}^\circ_{\mathsf{pk}}(c_\circ, d_\circ) \begin{cases} \perp, & \text{if } \mathsf{Open}_{\mathsf{pk}}(c, d) = \perp, \\ x, & \text{if } x = \mathsf{Open}_{\mathsf{pk}}(c, d) \wedge b = 0, \\ g(x, c_\circ), & \text{if } x = \mathsf{Open}_{\mathsf{pk}}(c, d) \wedge b = 1, \end{cases}$$

i.e., setting an evil bit b allows to manipulate commitments. Let us denote such a commitment scheme by $\mathcal{C}om^g$.

Theorem 2. *Let $\mathcal{C}om$ be (t, ε_1)-binding and (t, ε_2)-hiding. Then $\mathcal{C}om^g$ is (τ, ε_1)-binding and (τ, ε_2)-hiding where $\tau = t - \mathcal{O}(1)$. If $\mathcal{C}om$ is (t, ε_3)-extractable then $\mathcal{C}om^g$ is also (τ, ε_3)-extractable. If $\mathcal{C}om$ is (t, ε_4)-equivocable then $\mathcal{C}om^g$ is also (τ, ε_4)-equivocable.*

Proof. The proof is straightforward. Appendix B contains the proof and more detailed discussion of separation results and their implications. □

Corollary 1. *Let f be a combiner such that there exists efficiently computable functions g_a and g_b so that $f(r_a, g_b(r_b, c)) \oplus f(g_a(r_a, c), r_b) = 1$. Then the MA-3 scheme is insecure if we use h, f and g_a-malleable commitment scheme $\mathcal{C}om^{g_a}$.*

Proof. Recall that a protocol is insecure if for some valid input pairs protocol fails with probability 1. Let Alice's input be the first half of m_0 and Bob's input the second half of m_1. Then Charlie alters messages so that at the end Alice obtains m_0 and Bob m_1. Given a commitment $(0, c)$, Charlie forwards $(1, c)$ to Bob and sends $g_b(r_b, c)$ directly to Alice. Charlie forwards decommitment value d to Bob, who obtains $r'_a = g_a(r_a, c)$. Alice and Bob accept outputs m_0 and m_1 since $h(m_0, k) = h(m_1, k \oplus 1)$. □

Corollary 2. *If the protocol uses XOR combiner $f(x, y) = x \oplus y$ then an ordinary binding and hiding commitment is not sufficient for security.*

Proof. Obviously, since $f(r_a, r_b) \oplus f(r_a \oplus 1, r_b) = 1$. The latter shows that sending r_b in a scrambled way does not help as $g_b(r, c) = r$. □

We could not find a constructive counterexample for general class of combiners instead we used oracle separation to eliminate possibility of black-box proofs.

Theorem 3. *Let f be a left-right uniform combiner and $\mathcal{C}om$ a commitment scheme that remains hiding and binding commitment even if $f(\cdot, r)$ can be efficiently inverted for all $r \in \mathcal{K}$. Then there exists an oracle world where the MA-3 protocol is insecure, but $\mathcal{C}om$ is hiding and binding.*

Proof. CONSTRUCTION. Consider an oracle world where the oracle $\mathcal{O}$: (a) registers honest commitments; (b) creates inaccessible random commitments; (c) finds a "safe" solution to $f(r_a, r'_b) \oplus f(r'_a, r_b) = 1$. First, the commitment rule is modified so that every time an honest party computes $\mathsf{Com}_{\mathsf{pk}}(x, s)$ he also submits a tuple $(c, d, x, \perp, \perp, \perp)$

to $\mathcal{O}$. Secondly, $\mathcal{O}$ realises random transformations: given a commitment c, it looks whether c is already stored. If not returns $\perp$, otherwise $\mathcal{O}$ generates a random commitment $(c', d') \leftarrow \mathsf{Com}_{\mathsf{pk}}(x', s),\ s \leftarrow \mathcal{R}, x' \leftarrow \mathcal{K}$, updates tuple to (c, d, x, c', d', x') and outputs c'. Given a pair (c, d), $\mathcal{O}$ looks for a tuple (c, d, c', d') and if found outputs d'. Finally, given a *single* special call (c, r_b) oracle looks for a tuple (c, d, x, c', d', x'), if found finds r'_b such that $f(x, r'_b) \oplus f(x', r_b) = 1$, after that all such calls are ignored. INSECURITY. After Alice has sent c, Charlie submits c to $\mathcal{O}$ and forwards reply c' to Bob. After Bob has sent r_b, Charlie submits (c, r_b) to $\mathcal{O}$ and forwards r'_b to Alice. After Alice has sent d, Charlie sends (c, d) to $\mathcal{O}$ and forwards answer d' to Bob. Since $f(r_a, r'_b) = f(r'_a, r_b) \oplus 1$ Charlie has succeeded in deception. HIDING AND BINDING. We have to show that in the oracle world the modified $\mathcal{C}om$ is still hiding and binding. Binding is straightforward—we can perfectly simulate $\mathcal{O}$ as honest $\mathsf{Com}_{\mathsf{pk}}$ calls provide a decommitment and we can use inversion oracle for $f(\cdot, r)$. Thus, the advantage remains, only the working time increases to $\mathcal{O}(t \log t)$, since we have to manage a table of tuples. Lets establish that hiding is also preserved. Assume that $\mathcal{C}om$ is (τ, ε_1)-hiding and (τ, ε_2)-binding in the plain model where $\tau = \mathcal{O}(t \log t)$ is large enough but a t time adversary A achieves $\mathrm{Adv}^{\mathsf{hid}}_{\mathcal{O}}(A) > 4\varepsilon_1 + \varepsilon_2$ in the oracle world. Let B be the adversary that runs like A except B halts with 0, if A submits a *valid* decommitment (c_b, d) for the challenge c_b. The probability of early abort must be below $2\varepsilon_1 + \varepsilon_2$, otherwise A can either win the hiding game in the world with the inversion oracle or the complete simulation of hiding game provides enough double openings. The term $2\varepsilon_1$ comes from the fact that c' and r'_b together might leak some information about c_b and we have to use similar hybrid argument as demonstrated below, i.e., if substitute $\mathcal{O}$ with $\mathcal{O}'$ then the probability of early abort can drop by ε_1. Let $\mathcal{O}'$ do as $\mathcal{O}$ except that the update step for challenge c_b is different: given (c_b, d_b), the oracle $\mathcal{O}'$ computes $(c'', d'') \leftarrow \mathsf{Com}_{\mathsf{pk}}(x'', s),\ s \leftarrow \mathcal{R}, x', x'' \leftarrow \mathcal{K}$, updates tuple to (c, d, x, c'', d'', x') and outputs c''. Clearly, $\mathrm{Adv}^{\mathsf{hid}}_{\mathcal{O}}(B) - \mathrm{Adv}^{\mathsf{hid}}_{\mathcal{O}'}(B) \leq \varepsilon_1$ or otherwise we can use the simulation of the oracle world to win the true hiding game. Since c' and r'_b are completely independent and r'_b has uniform distribution as f is left-right uniform, we can perfectly simulate interaction of $\mathcal{O}'$ and B in the plain model. A contradiction $\mathrm{Adv}^{\mathsf{hid}}_{\mathcal{O}'}(B) > \varepsilon_1$. □

Theorem 3 shows that there are no black-box security proofs of MA-3 that assume only binding and hiding from commitment scheme, i.e. we have to assume some kind of non-malleability of $\mathcal{C}om$ or provide explicit white-box security proof for an instantiation of the MA-3.

4.2 The SAS protocol

Note that the SAS protocol requires non-malleable commitment as we can define function $g : \mathcal{M} \to \mathcal{M}$ so that $g(m_0 || r_a) = m_1 || r_a$. Vaudenay assumed that commitment returns only r_a and m_a is explicitly included in the decommitment value d but clearly this is a cosmetic difference.

Theorem 4. *Let $\mathcal{C}om^g$ be a g-malleable but (t, ε)-hiding and (t, ε)-binding commitment scheme. Then the SAS protocol is insecure.*

Proof. Recall that protocol is insecure if for some valid input protocol fails with probability 1. Let Alice's input be m_0. Then given a commitment $(0, c)$, Charlie forwards $(1, c)$ to Bob and sends r_b directly to Alice. Charlie forwards decommitment value d to Bob, who obtains $m_1||r_a$. Bob accepts outputs m_1 since he got the correct r_a. □

Corollary 3. *Extractability or equivocability are not sufficient to guarantee security of the SAS protocol.* □

Results indicate that the proofs of Theorem 5 of [Vau05] are incorrect. Indeed, in the case of extractable commitments sk is used to win the hiding game that is absurd, since after the secret key sk has been leaked there is no privacy guarantees. Similarly, it does not make sense to use sk and faked commitments to beat the binding game, since a leakage of sk removes binding guarantees. Nevertheless, the original proof is valid if one assumes CCA security from the commitment scheme, since the proof actually uses calls to decommitment oracle to win the hiding and binding games. On the other hand, CCA security implies non-malleability. Use of more advanced combiner $f(m_a, r_a, r_b)$ does not alleviate the security requirements as it is equivalent to MA-3 with empty m_b.

5 Security proofs

Let forge denote the event that the adversary A succeeds, i.e. Alice and Bob have coinciding `check` values but $m_a||m'_b \neq m'_a||m_b$. Then the advantage of A is defined as

$$\mathrm{Adv}^{\mathsf{msg\text{-}for}}(A) = \max_{m_a, m_b} \Pr\left[\mathsf{forge}\right] \enspace .$$

An authentication protocol is (t, ε)-secure in the stand-alone model if for any t time adversary A, we have $\mathrm{Adv}^{\mathsf{msg\text{-}for}}(A) \leq \varepsilon$. As both protocols are asynchronous, the adversary can deliver messages before they are sent. Denote by $\mathsf{send}(i)$ that the ith message was sent and $\mathsf{recv}(i)$ that the ith message was received by honest parties. Then causal relations $\mathsf{send}(1) \prec \mathsf{recv}(2) \prec \mathsf{send}(3)$ and $\mathsf{recv}(1) \prec \mathsf{send}(2) \prec \mathsf{recv}(3)$ still hold. In the following we divide execution paths to classes. An execution path is almost normal (denoted as norm) if execution second round is completed before the third round is started

$$\mathsf{recv}(1), \mathsf{recv}(2), \mathsf{send}(1), \mathsf{send}(2) \prec \mathsf{send}(3) \enspace ,$$
$$\mathsf{recv}(1), \mathsf{recv}(2), \mathsf{send}(1), \mathsf{send}(2) \prec \mathsf{recv}(3) \enspace .$$

An execution path is abnormal (denoted as $\neg\mathsf{norm}$) if one of the conditions is violated, i.e., one of the mutually exclusive events

$$\mathsf{send}(3) \prec \mathsf{send}(2) \qquad \text{or} \qquad \mathsf{recv}(3) \prec \mathsf{recv}(2) \tag{1}$$

happens. Further analysis shows that abnormal executions fail with high probability provided that the commitment scheme is hiding and binding. Almost normal execution is secure under more restrictive assumptions.

Lemma 1. *For any t there exists $\tau = t + \mathcal{O}(1)$ such that if $\mathcal{C}om$ is (τ, ε_1)-hiding, f is (τ, ε_5)-pseudorandom and h a uniform hash function. Then for any t time adversary C*

$$\Pr\left[\mathsf{forge}_{\mathsf{sas}} \wedge \mathsf{recv}(3) \prec \mathsf{recv}(2)\right] \le 2^{-\ell} \cdot \Pr\left[\mathsf{recv}(3) \prec \mathsf{recv}(2)\right] + \varepsilon_1$$
$$\Pr\left[\mathsf{forge}_{\mathsf{ma3}} \wedge \mathsf{recv}(3) \prec \mathsf{recv}(2)\right] \le 2^{-\ell} \cdot \Pr\left[\mathsf{recv}(3) \prec \mathsf{recv}(2)\right] + \varepsilon_1 + \varepsilon_5$$

for the SAS and the MA-3 protocols.

Proof. Since $\mathsf{send}(2) \prec \mathsf{recv}(3) \prec \mathsf{recv}(2) \prec \mathsf{send}(3)$, then values c', r_b, d', r_b' are fixed before the adversary sees a decommitment value d. In particular, $\mathtt{check}_b$ is also fixed before $\mathsf{send}(3)$ and C succeeds if he guesses the value of c. More formally, we can convert C into a distinguisher A (the next construction is for the SAS protocol):

1. Choose $r_a \leftarrow \mathcal{K}$ and send $m_a || r_a$ as a challenge x_0.
2. Given c_b simulate protocol until $\mathsf{recv}(2)$. If $\mathsf{recv}(2) \prec \mathsf{recv}(3)$ then halt with 0.
3. Compute $\mathtt{check}_a = r_a \oplus r_b'$ and $\mathtt{check}_b = r_a' \oplus r_b$ output 1 iff they coincide.

If $b = 0$ then protocol is perfectly simulated and A outputs 1 with the probability $\Pr\left[\mathsf{forge} \wedge \mathsf{recv}(3) \prec \mathsf{recv}(2)\right]$. If $b = 1$ then the protocol run is independent of r_a and $\Pr\left[r_a \oplus r_b' = r_a' \oplus r_b\right] = 2^{-\ell}$. As a result we get

$$\mathrm{Adv}^{\mathsf{hid}}(A) \ge \Pr\left[\mathsf{forge} \wedge \mathsf{recv}(3) \prec \mathsf{recv}(2)\right] - 2^{-\ell} \cdot \Pr\left[\mathsf{recv}(3) \prec \mathsf{recv}(2)\right]$$

and the first claim follows. For the MA-3 protocol the construction is exactly the same, except $x_0 = r_a$ and $\mathtt{check}$ values are computed differently. Similarly, r_a is independent from the simulated protocol run when $b = 1$. Since f is (τ, ε_5)-pseudorandom the $\mathtt{check}_a$ is also $(\tau - \mathcal{O}(1), \varepsilon_5)$-pseudorandom and hence the second claim follows. □

Lemma 2. *Assume that A is a t-time adversary and let f be a right-uniform combiner and h a uniform hash function. If $\mathcal{C}om$ is perfectly binding, then for both protocols*

$$\Pr\left[\mathsf{forge} \wedge \mathsf{send}(3) \prec \mathsf{send}(2)\right] \le 2^{-\ell} \cdot \Pr\left[\mathsf{send}(3) \prec \mathsf{send}(2)\right] \quad .$$

Otherwise $\mathcal{C}om$ must be (τ, ε_2)-binding with $\tau = \frac{56\alpha t}{\varepsilon_2'(1-\varepsilon_2)}$ for a constant [6] α to assure

$$\Pr\left[\mathsf{forge} \wedge \mathsf{send}(3) \prec \mathsf{send}(2)\right] \le 2^{-\ell} \cdot \Pr\left[\mathsf{send}(3) \prec \mathsf{send}(2)\right] + \varepsilon_2'$$

where $\varepsilon_2' > 4 \cdot 2^{-\ell}$ for the SAS and $\varepsilon_2' > 4 \cdot |\mathcal{K}_b|^{-1}$ for the MA-3 protocol.

Proof. Since $\mathsf{recv}(2) \prec \mathsf{send}(3) \prec \mathsf{send}(2)$ then m_a', c', m_b', r_b' is sent before r_b, hence $\mathtt{check}_a$ is fixed before the adversary sees r_b. Similarly to the proof of Lemma 1, choosing r_a' independently from r_b leads to a success probability $2^{-\ell}$ and the first claim follows. For general case, the adversary must double open commitments to achieve a better success. Still, we need replies r_b^0 and r_b^0 such that adversary opens the commitment c' differently. Consider a matrix $H[s, r_b]$ with columns $r_b \in \mathcal{K}_b$ and rows $s \in \mathcal{R}$ capturing

[6] The small constant α comes from the overhead of Damgård-Fujisaki knowledge extractor (Appendix C): the procedure has to re-initialise the protocol after each probe and do some local bookkeeping.

all other random bits of the protocol including also the Gen algorithm. Set $H[s, r_b] = 1$ iff $\mathsf{check}_a = \mathsf{check}_b$. In case of the SAS protocol, the matrix H is complete as for each r_b there is a single suitable r'_a. However in the MA-3 protocol, two key values $f(r'_a, r^0_b)$ and $f(r'_a, r^0_b)$ can lead to same check_b. To eliminate false positives, we store the first successful open value $r'_a[s]$ and test whether another successful deception leads to different r'_a. Alternatively stated, we dynamically set all other row elements $H[s, r_b]$ leading to $r'_a[s]$ to zero. Since f is right-uniform and h is a uniform hash function, there is $2^{-\ell} \cdot |\mathcal{K}|$ keys that correspond to check_a. As a result the effective probability

$$\begin{aligned}\varepsilon &= \Pr\left[s \leftarrow \mathcal{R}, r_b \leftarrow \mathcal{K} : H[s, r_b] = 1\right] \\ &\geq \Pr\left[\mathsf{forge} \wedge \mathsf{send}(3) \prec \mathsf{send}(2)\right] - 2^{-\ell} \cdot \Pr\left[\mathsf{send}(3) \prec \mathsf{send}(2)\right] \geq \varepsilon'_2\end{aligned}$$

for the MA-3 protocol. Corollary 4 assures that there is a $\frac{56\alpha t}{\varepsilon'_2(1-\varepsilon_2)}$-time probing algorithm that finds r^0_b, r^1_b corresponding to double opening of c' with success ε_2. □

Theorem 5. *Let t be a desired time bound. Let $\mathcal{C}om$ be (τ_2, ε_2)-binding. If $\mathcal{C}om$ is perfectly binding set $\varepsilon'_2 = 0$, otherwise set $\varepsilon'_2 = \max\{4 \cdot 2^{-\ell}, 56\alpha/(\tau_2(1-\varepsilon_2))\}$ where α is a known small constant. Then there exist $\tau_1 = t + \mathcal{O}(1)$ and $\tau_3 = \mathcal{O}(t)$ such that if $\mathcal{C}om$ is (τ_1, ε_1)-hiding and (τ_3, ε_3)-non-malleable commitment scheme then the SAS protocol is $(t, 2^{-\ell} + \varepsilon_1 + \varepsilon'_2 + \varepsilon_3)$-secure.*

Proof. Let C be a malicious environment that achieves $\mathrm{Adv}^{\text{msg-for}}(C) > 2^{-\ell} + \varepsilon_1 + \varepsilon'_2 + \varepsilon_3$ and let m_a be the corresponding input. We build an adversary $A = (A_1, A_2, A_3, A_4)$ that can break non-malleability of the commitment scheme. A_1 outputs a description of uniform distribution over $\{m_a\} \times \mathcal{K}$ and $\sigma_1 = (\mathsf{pk}, m_a)$. Given c, σ_1, A_2 simulates the protocol with $r_b \leftarrow \mathcal{K}$ and stops before $\mathsf{send}(3)$. In case of abnormal execution (1) or $c = c'$, A halts with 0. Otherwise, A_2 outputs a commitment c' and σ_2 containing enough information to resume the simulation and (r_b, r'_b). Given d, σ_2, A_3 resumes the simulation and outputs d' as a decommitment value. If A_3 was successful in opening then A_4 gets $x = m_a||r_a$, $y_1 = m'_a||r'_a$ and σ_2 containing (r_b, r'_b). A_4 computes two check values $\mathsf{check}_a = r_a \oplus r'_b$ and $\mathsf{check}_b = r'_a \oplus r_b$ and outputs 1 if $\mathsf{check}_a = \mathsf{check}_b$. As only abnormal execution, $c = c'$ or a protocol failure $\mathsf{Open}_{\mathsf{pk}}(c', d') = \perp$ causes a premature halting of A, we get

$$\begin{aligned}\Pr\left[A_4 = 1|\mathsf{World}_0\right] &= \Pr\left[\mathsf{forge} \wedge \mathsf{norm}\right] \quad , \\ \Pr\left[A_4 = 1|\mathsf{World}_1\right] &\leq 2^{-\ell} \cdot \Pr\left[\mathsf{norm}\right] \quad .\end{aligned}$$

Combining the result with Lemmas 1 and 2, we get a desired contradiction

$$\begin{aligned}\Pr\left[\mathsf{forge} \wedge \mathsf{norm}\right] &\geq \mathrm{Adv}^{\text{msg-for}}(C) - 2^{-\ell} \cdot \Pr\left[\neg\mathsf{norm}\right] - \varepsilon_1 - \varepsilon'_2 \quad , \\ \mathrm{Adv}^{\text{nm}}(A) &\geq \mathrm{Adv}^{\text{msg-for}}(C) - 2^{-\ell} - \varepsilon_1 - \varepsilon'_2 > \varepsilon_3 \quad .\end{aligned}$$

□

Theorem 6. *Let t be a desired time bound. Let $\mathcal{C}om$ be (τ_2, ε_2)-binding. If $\mathcal{C}om$ is perfectly binding set $\varepsilon'_2 = 0$, otherwise set $\varepsilon'_2 = \max\{4 \cdot |\mathcal{K}_b|^{-1}, 56\alpha/(\tau_2(1-\varepsilon_2))\}$*

where α is a known small constant. Then there exist $\tau_1 = t + \mathcal{O}(1)$ and $\tau_3 = \mathcal{O}(t)$ such that if $\mathcal{C}om$ is (τ_1, ε_1)-hiding and (τ_3, ε_3)-non-malleable commitment scheme, h is ε_4-almost universal uniform hash function and f is (τ_1, ε_5)-pseudorandom permutation, then MA-3 protocol is $(t, 2^{-\ell} + 2\varepsilon_1 + \varepsilon_2' + \varepsilon_3 + \varepsilon_4 + 3\varepsilon_5)$-secure.

Proof. Let C be a malicious environment that achieves $\mathrm{Adv}^{\textsf{msg-for}}(C) > 2^{-\ell} + 2\varepsilon_1 + \varepsilon_2' + \varepsilon_3 + \varepsilon_4 + 3\varepsilon_5$ and let m_a, m_b be the corresponding input. We build an adversary $A = (A_1, A_2, A_3, A_4)$ that can break non-malleability of the commitment scheme. A_1 outputs a description of uniform distribution over $\mathcal{K}_a$ and $\sigma_1 = (\mathsf{pk}, m_a, m_b)$. Given c, σ_1, A_2 simulates the protocol with $r_b \leftarrow \mathcal{K}_b$ and stops before $\mathsf{send}(3)$. In case of abnormal execution (1) or $c = c'$, A halts with 0. Otherwise, A_2 outputs a commitment c' and σ_2 containing enough information to resume the simulation and $(m_a, m_a', m_b, m_b', r_b, r_b')$. Given d, σ_2, A_3 resumes the simulation and outputs d' as a decommitment value. If A_3 was successful in opening then A_4 gets $x = r_a$, $y_1 = r_a'$ and σ_2 containing $(m_a, m_a', m_b, m_b', r_b, r_b')$. A_4 computes check values $\mathsf{check}_a = h(m_a || m_b', f(r_a, r_b'))$ and $\mathsf{check}_b = h(m_a' || m_b, f(r_a', r_b))$ and outputs 1 if $\mathsf{check}_a = \mathsf{check}_b$. Since only abnormal execution, $c = c'$ or a protocol failure $\mathsf{Open}_{\mathsf{pk}}(c', d') = \perp$ causes a premature halting of A, we get

$$\Pr\left[A_4 = 1 | \mathsf{World}_0\right] = \Pr\left[\mathsf{forge} \wedge \mathsf{norm} \wedge c \neq c'\right] \ ,$$
$$\Pr\left[A_4 = 1 | \mathsf{World}_1\right] \leq 2^{-\ell} \cdot \Pr\left[\mathsf{norm} \wedge c \neq c'\right] + \varepsilon_5 \ ,$$

as for random key k, the control value $h(m_a || m_b', k)$ has uniform distribution and f is a (τ_1, ε_5)-pseudorandom function. Since $\mathcal{C}om$ is (τ_1, ε_1)-hiding and f pseudorandom and h is ε_4-almost universal, we get by hybrid argument

$$\Pr\left[\mathsf{forge} \wedge \mathsf{norm} \wedge c = c' \wedge r_b' = r_b\right] \leq \varepsilon_1 + \varepsilon_5 + \varepsilon_4 \ ,$$
$$\Pr\left[\mathsf{forge} \wedge \mathsf{norm} \wedge c = c' \wedge r_b' \neq r_b\right] \leq \varepsilon_1 + \varepsilon_5 + 2^{-\ell} \cdot \Pr\left[\mathsf{norm} \wedge c = c'\right] \ ,$$
$$\Pr\left[\mathsf{forge} \wedge \mathsf{norm} \wedge c = c'\right] \leq \varepsilon_1 + \varepsilon_5 + \varepsilon_4 + 2^{-\ell} \cdot \Pr\left[\mathsf{norm} \wedge c = c'\right] \ ,$$

where in the third inequality corresponds to more precise combined hybrid argument. Combining the results with Lemmas 1 and 2 and we get a desired contradiction

$$\Pr\left[\mathsf{forge} \wedge \mathsf{norm}\right] \geq \mathrm{Adv}^{\textsf{msg-for}}(C) - 2^{-\ell} \cdot \Pr\left[\neg \mathsf{norm}\right] - \varepsilon_1 - \varepsilon_2' - \varepsilon_5 \ ,$$
$$\mathrm{Adv}^{\textsf{nm}}(A) \geq \mathrm{Adv}^{\textsf{msg-for}}(C) - 2^{-\ell} - 2\varepsilon_1 - \varepsilon_2' - \varepsilon_4 - 3\varepsilon_5 > \varepsilon_3 \ .$$

□

Remark 1. Note that in all proofs we needed that f is (τ_1, ε_5)-pseudorandom permutation only if adversary can query at most two values of f.

6 Suggested implementation

To implement the protocol, one has to fix a hash function, a non-malleable commitment scheme and good pseudorandom combiner. For the commitment scheme there are essentially two alternatives either we use relatively slow asymmetric primitives or relay on fast symmetric cryptography. The choice is not a clear-cut and depends on desired security goals. In a nutshell, asymmetric methods provide provable high level security that might be considered unnecessary as total failure probability is above $2^{-\ell}$.

Example construction of commitment schemes. The simplest construction of a non-malleable commitment scheme is based on a CCA2 secure encryption scheme. Let $\mathsf{Enc}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ be deterministic encryption rule where $r \in \mathcal{R}$ denotes randomness used to encrypt a message. Define $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x, r)$ as $c = \mathsf{Enc}_{\mathsf{pk}}(x, r)$ and $d = (x, r)$ and $\mathsf{Open}_{\mathsf{pk}}(c, d) = m$ if $\mathsf{Enc}_{\mathsf{pk}}(x, r) = c$ and $\perp$ otherwise. Then the corresponding commitment scheme is CCA secure provided that pk is generated by a trusted party. For some encryption schemes participants can generate pk themselves. We suggest Cramer-Shoup encryption scheme [CS98] as the public key is a random triple of group elements and there is no need for trusted party. Nevertheless, both parties must have the same key pk since *a priory* non-malleability w.r.t. single public key does not guarantee non-malleability w.r.t. related keys pk_1 and pk_2. Another alternative is RSA-OAEP that is CCA secure in a random oracle model [FOPS01]. Even more *heuristic* way to construct non-malleable commitments is based on collision resistant hash functions. If we define $(c, d) \leftarrow \mathsf{Com}(x, r)$ with $c = h(x, r)$ and $d = (x, r)$, there are no guarantees for hiding. Nevertheless, if we use hash function with OAEP padding

$$c = h(s, t), \quad s = (x||0^{k_0}) \oplus G(r), \quad t = r \oplus H(s),$$

then the commitment scheme is provably hiding and binding in the random oracle model provided that h is collision resistant. The security proof [FOPS01] of the OAEP padding assumes that h is a partial-domain one-way permutation. More specifically, it should be infeasible to find s given $h(s, t),\ s \leftarrow \mathcal{D}_1, t \leftarrow \mathcal{D}_2$. The partial one-wayness follows from one-wayness provided that h is at least (t, ε)-one-way function with $|\mathcal{D}_2| \ll t$. The other assumption that h is a permutation is important in the proof. Therefore, we can only *conjecture* that the proof can be generalised and OAEP provides a CCA secure commitment scheme. Since hash-commitments are not perfectly-binding, Lemma 2 requires that the commitment is (τ, ε_2)-binding where τ is inversely proportional to ε_2'. The impact of ε_2 is irrelevant as long $\varepsilon_2 < 1/2$. Shortly put, birthday paradox assures that for k bit hash values $\varepsilon_2' \leq c \cdot 2^{-k/2} \cdot t^{-1},\ c < 1$ and therefore hash values must be quite long to get reasonable security guarantees.

Example constructions of combiners. Ideally, we should use left-right uniform combiner that is also pseudorandom w.r.t. 2 calls of f. If we set $\mathcal{K}_a = \{0,1\}^{2m}$ and $\mathcal{K}_b = \{0,1\}^m$ then the most natural combiner $f(x_0||x_1, y) = x_0 y + x_1$ over the Galois field $\mathsf{GF}(2^m)$ is indeed $(\infty, 0)$-pseudorandom. On the other hand, r_a is twice as long as the hash key. As commitments are the computational bottleneck of the protocol, an appealing alternative is to use AES with 128-bit key to compute $f(x_1||\ldots||x_s, y_1||\ldots||y_s) = AES(x_1, y_1)||\ldots||AES(x_s, y_s)$ on 128-bit blocks.

Example constructions of hash families Constructions of ε-AU$_2$ hash families have the property that the value of ε depends on the length of message inputs. In our application the tag space $\mathcal{T}$ is relatively small and it is desired that $\varepsilon \approx 1/|\mathcal{T}|$. As shown in [BJKS05] the effect of message length can be eliminated using constructions based on concatenation of hash families. Towards this end the following composition theorem from [Sti92] is useful.

Theorem 7. *If there exists an ε_1-AU$_2$ hash family $\mathcal{H}_1 = \{f : \mathcal{D} \to \mathcal{T}_1\}$ and an ε_2-AU$_2$ hash family $\mathcal{H}_2 = \{g : \mathcal{T}_1 \to \mathcal{T}_2\}$, then there exist an ε-AU$_2$ hash family $\mathcal{H}$ of hash functions from $\mathcal{D}$ to $\mathcal{T}_2$, where $\varepsilon \leq \varepsilon_1 + \varepsilon_2$. If, moreover, the hash functions of the second family $\mathcal{H}_2$ are uniform, then also the hash functions of the family $\mathcal{H}$ are uniform.*

The hash functions in $\mathcal{H}$ are constructed as composed functions of hash function in $\mathcal{H}_1$ and $\mathcal{H}_2$. Let ℓ be the length of the final tag in bits, and assume that all messages in the set $\mathcal{D}$ have at most 2^ℓ blocks of 2ℓ bits. Then we can construct an $2^{-\ell}$-AU$_2$ hash family $\mathcal{H}_1$ from the message space $\mathcal{D}$ and with tag length of 2ℓ bits as follows. Let $\mathsf{GF}(2^{2\ell})$ be a Galois field of $2^{2\ell}$ elements. We denote by $x_0||x_1||\ldots||x_{m-1}$ the message blocks of x, where $x_i \in \mathsf{GF}(2^{2\ell})$ and $m < 2^\ell$. For $k_1 \in \mathcal{K}_1 = \mathsf{GF}(2^{2\ell})$ we set

$$f_{k_1}(x) = x_{m-1}k_1^{m-1} + \ldots + x_1 k_1 + x_0 \text{ over } \mathsf{GF}(2^{2\ell}) \ .$$

Then it can be shown that the family $\mathcal{H}_1 = \{f_{k_1}\}$ is $2^{-\ell}$-AU$_2$ hash family. The second hash family $\mathcal{H}_2$ is defined similarly with message space $\mathsf{GF}(2^{2\ell})$, with the key space $\mathcal{K}_2 = \mathsf{GF}(2^\ell)$, and with the tag space $\mathcal{T} = \mathsf{GF}(2^\ell)$. The family $\mathcal{H}_2$ consists of all functions g_{k_2} of the form

$$g_{k_2}(y_0||y_1) = y_1 k_2 + y_0 \text{ over } \mathsf{GF}(2^\ell) \ .$$

The family $\mathcal{H}_2$ is an $2^{-\ell+1}$-AU$_2$ hash family. By Theorem 7 the family $\mathcal{H}$ consisting of hash functions $h_{k_1,k_2} = g_{k_2} \circ f_{k_1}$ is then ε-AU$_2$ hash family with $\varepsilon = 2^{-\ell+2}$, and key space $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2$ consisting of strings of 3ℓ bits.

References

[BJKS05] J. Bierbrauer, T. Johansson, G. Kabatianskii, and B. Smeets. On families of hash functions via geometric codes and concatenation. In *D. Stinson (Ed.): CRYPTO '93, LNCS 773*, 2005.

[BM92] S. Bellovin and M. Merrit. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California*, May 1992.

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *M. Wiener (Ed.)Advances in Cryptology – CRYPTO '99, LNCS 1666*, pages 519–536, 1999.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *STOC '98*, pages 141–150, 1998.

[CKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In *B. Pfitzmann (Ed.) EUROCRYPT 2001, LNCS 2045*, pages 40–59, 2001.

[Cre02] Giovanni Di Crescenzo. Equivocable and extractable commitment schemes. In *SCN 2002*, pages 74–87, 2002.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *H. Krawzcyk (Ed.) CRYPTO '98, LNCS 1462*, pages 13–25, 1998.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *STOC '91*, pages 542–552, New York, NY, USA, 1991. ACM Press.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *C. Boyd (Ed.) ASIACRYPT 2002, LNCS 2248*, pages 125–142, 2002.

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC 2003*, pages 426–437, 2003.

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In *M. Bellare (Ed.) CRYPTO 2000, LNCS 1880*, pages 413–431, 2000.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *B. Kaliski (Ed.) CRYPTO '97*, pages 16–30, 1997.

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. Rsa-oaep is secure under the rsa assumption. In *J. Kilian (Ed.), CRYPTO 2001, LNCS 2139*, pages 260–274, 2001.

[GMN04] Christian Gehrmann, Chris J. Mitchell, and Kaisa Nyberg. Manual authentication for wireless devices. *RSA Cryptobytes*, 7(1):29–37, January 2004.

[Gol04] Oded Goldreich. *Foundations of Cryptography*, volume 2. Cambridge University Press, 2004.

[GPS04] Christian Gehrmann, Joakim Persson, and Ben Smeets, editors. *Bluetooth Security*, chapter 9.1.2: Improved pairing, pages 140–149. Artech House, 2004.

[Hoe05] Jaap-Henk Hoepman. Ephemeral pairing on anonymous networks. In *SPC 2005*, pages 101–116, 2005.

[Hun05] Preston Hunt. Wireless USB Association Models. Presentation at the Wireless USB Developers Conference, May 2005. Available from `http://www.usb.org/developers/wusb/docs/presentations/`.

[INQ05] INQUIRER staff. Intel, Microsoft Vista promise Wi-Fi Simple Config. News article in "the INQUIRER", August 2005. `http://www.theinquirer.net/?article=25777`.

[KOY01] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords . In *B. Pfitzmann (Ed.), EUROCRYPT 2001, LNCS 2045)*, 2001.

[KW04] Amol Kulkarni and Jesse Walker. Easy and secure setup of personal wireless networks. *Technology@Intel Magazine*, November 2004. `http://www.intel.com/technology/magazine/communications/wi11042.pdf`.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *J. Feigenbaum (Ed.) CRYPTO 1991*, pages 129–140, 1991.

[Sar80] Dilip V. Sarwate. A note on universal classes of hash functions. *Inf. Process. Lett.*, 10(1):41–45, 1980.

[SCP00] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Necessary and sufficient assumptions for non-iterative zero-knowledge proofs of knowledge for all np relations. In *ICALP 2000*, pages 451–462, 2000. Extended version was also published in SIAM J. Comp. 30(2).

[Sti92] D. R. Stinson. Universal Hashing and Authentication Codes. In *J. Feigenbaum (Ed.), Advances in Cryptology - CRYPTO '91*, 1992.

[Vau05] Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *V. Shoup (Ed.), CRYPTO 2005, LNCS 3621*, pages 309–326, 2005.

A CCA security implies non-malleability

Theorem 8. *Let $\mathcal{C}om$ be (t, ε_1)-hiding and (t, ε_2)-binding under chosen commitment attack. If $\mathcal{C}om$ is perfectly binding, then $\mathcal{C}om$ is $(\tau, 2\varepsilon_1)$-non-malleable, otherwise, $\mathcal{C}om$ is (τ, ε_2)-non-malleable for $\tau = 2t + \mathcal{O}(1)$.*

Proof. PERFECTLY-BINDING. First ε-security against real-or-random game implies security against left-or-right game where the adversary outputs both x_0 and x_1. Now consider an adversary $A = (A_1, A_2, A_3, A_4)$ that has $\mathrm{Adv}^{\mathsf{nm}}(A) > 2\varepsilon$. Then B chooses $x_0, x_1 \leftarrow \mathcal{M}_0$. Now given c_b, B simulates the game until A_2 outputs σ_2 and $(c_1, \dots, c_n)$. Use decommitment oracle to find corresponding vector $(y_1, \dots, y_n)$. Output the end result of $A_4(x_0, y_1, \dots, y_n, \sigma_2)$. Clearly, we provide perfect simulation World_b, except we do not terminate when A_3 fails. Fortunately, the latter term cancels out due to the symmetry and $\mathrm{Adv}^{\mathsf{hid}}_{\mathcal{O}_{\mathsf{dec}}}(B) = \mathrm{Adv}^{\mathsf{nm}}(A)$. NON-BINDING CASE. Let A be the contradicting adversary $\mathrm{Adv}^{\mathsf{bind}}_{\mathcal{O}_{\mathsf{dec}}}(A) > \varepsilon_2$. Then B chooses $x_0, x_1 \leftarrow \mathcal{M}_0$ and sets $c \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_0, s),\ s \leftarrow \mathcal{R}$ and simulates the non-malleability game. After A_2 has stopped, B queries decommitments d_0, d_1 of c to x_0 and x_1. As A_3 must succeed with probability at least $\mathrm{Adv}^{\mathsf{nm}}(A)$ we get a double opening with probability $\mathrm{Adv}^{\mathsf{nm}}(A) > \varepsilon_2$. □

B Separation results

Theorem 2. *Let $\mathcal{C}om$ be (t, ε_1)-binding and (t, ε_2)-hiding. Then $\mathcal{C}om^g$ is (τ, ε_1)-binding and (τ, ε_2)-hiding where $\tau = t - \mathcal{O}(1)$. If $\mathcal{C}om$ is (t, ε_3)-extractable then $\mathcal{C}om^g$ is also (τ, ε_3)-extractable. If $\mathcal{C}om$ is (t, ε_4)-equivocable then $\mathcal{C}om^g$ is also (τ, ε_4)-equivocable.*

Proof. HIDING AND BINDING. Adding an extra 0 before the commitment cannot decrease indistinguishability. Double opening w.r.t. $\mathcal{C}om^g$ must produce a valid double opening $\mathsf{Open}_{\mathsf{pk}}(c, d_0) \neq \mathsf{Open}_{\mathsf{pk}}(c, d_1)$ regardless whether $b = 0, 1$.EXTRACTABILITY. As all commitments are in the form (b, c) then defining $\mathsf{Extr}^{\circ}_{\mathsf{sk}}(0, c) = \mathsf{Extr}_{\mathsf{sk}}(c)$ and $\mathsf{Extr}^{\circ}_{\mathsf{sk}}(1, c) = g(\mathsf{Extr}_{\mathsf{sk}}(c))$ is sufficient. EQUIVOCABILITY. Define $\mathsf{Com}^{*\circ}_{\mathsf{sk}} = (0, c)$ where $c \leftarrow \mathsf{Com}^{*}_{\mathsf{sk}}$ and $\mathsf{Equiv}^{\circ}_{\mathsf{sk}} = \mathsf{Equiv}_{\mathsf{sk}}$. □

A more natural example of malleable commitments is following. Fix $\mathcal{C}^{\circ} = \mathcal{K} \times \mathcal{C}$ and define $\mathsf{Com}^{\circ}_{\mathsf{pk}} : \mathcal{K} \times (\mathcal{R} \times \mathcal{K}) \rightarrow \mathcal{C}^{\circ} \times \mathcal{D}$ as $(c_{\circ}, d) \leftarrow \mathsf{Com}^{\circ}_{\mathsf{pk}}(x, s, y)$ where $c_{\circ} = (y, c)$ and $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x \oplus y, s)$. Define $\mathsf{Open}^{\circ}_{\mathsf{pk}}(c_{\circ}, d_{\circ}) = x \oplus y$ if $x = \mathsf{Open}_{\mathsf{pk}}(c, d) \neq \perp$. Then Theorem 2 still holds and provides more natural separation between non-malleability and other properties.

C A knowledge extraction lemma

Damgård and Fujisaki have developed a simple black-box knowledge extractor [DF02, App. A] that allows to lower bound probability of double openings in Lemma 2.

Lemma 3. *Let $H[s, r]$ with $s \in \mathcal{R}$ and $r \in \mathcal{K}$ be a binary matrix. Let the probability $\Pr\left[s \leftarrow \mathcal{R}, r \leftarrow \mathcal{K} : H[r, s] = 1\right] = \varepsilon > 4 \cdot |\mathcal{K}|^{-1}$. Then there exists a probabilistic probing strategy that finds $H[s, r_1] = H[s, r_2] = 1$ in expected time less than $\frac{56}{\varepsilon}$ probes.*

Corollary 4. *Let $H[s, r]$ with $s \in \mathcal{R}$ and $s \in \mathcal{K}$ be a binary matrix. Let the probability $\Pr\left[s \leftarrow \mathcal{R}, r \leftarrow \mathcal{K} H[s, r] = 1\right] = \varepsilon > 4 \cdot |\mathcal{K}|^{-1}$. Then there exists a probabilistic probing strategy that probes at most $\tau = \frac{56}{\varepsilon\delta}$ entries and fails to find $H[s, r_1] = H[s, r_2] = 1$ with probability at most δ.*

Proof. Follows directly from the Markov inequality. □