CYBERNETICA
Institute of Information Security

On the (Im)possibility of Perennial Message Recognition Protocols without Public-Key Cryptography

Madeline González Muñiz, Peeter Laud

T-4-12 / 2011

Madeline González Muñiz[1], Peeter Laud[1,2].
[1] Cybernetica, Institute of Information Security, [2] University of Tartu, Institute of Computer Science

The research reported here was supported by:

1. Estonian Science foundation, grant(s) No. 8124,
2. the target funded theme SF0012708s06 "Theoretical and Practical Security of Heterogenous Information Systems",
3. the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS

Cybernetica research reports are available online at
`http://research.cyber.ee/`

Mailing address:
AS Cybernetica
Akadeemia tee 21
12618 Tallinn
Estonia

On the (Im)possibility of Perennial Message Recognition Protocols without Public-Key Cryptography

Madeline González Muñiz, Peeter Laud

March 21, 2011

Abstract

A message recognition protocol (MRP) aims to exchange authenticated information in an insecure channel using resource-restricted devices. During the initialization session of the protocol, the parties exchange some authenticated information which the adversary can passively observe. Then, one party wants to send authenticated messages to the other party in an insecure channel. Such security requirements are often found in wireless sensor networks, where two nodes want to keep communicating after initially meeting each other, but where all communication can be observed by the adversary.

A common way for ensuring the authenticity is to initialize a public-key primitive (signature scheme) during the initialization session. However, the identity of the user does not need to be authenticated in our setting, so we do not need a certificate authority that binds keys to users. Another way to ensure authenticity is for the first party to commit to certain values during the initialization (typically by using hash chains), such that these values can be later opened and used to authenticate messages.

Public-key cryptography is computationally intensive. Protocols based on hash chains also have significant computational requirements, but more importantly, they are not *perennial* — the number of possible message authentications is fixed in the initialization phase. Although efficient perennial MRPs based on hashing have been proposed, they have been shown to be flawed.

In this paper, we show that if we restrict the primitives our protocol can use to a set that is usually understood to comprise "symmetric cryptography", then it is impossible to construct a perennial MRP. We show that, without a common secret, authenticating a message is not possible. We also show that a common secret cannot be agreed on during the initialization session. Our result should be an interesting guideline for authentication protocols in general, showing that initial authentication cannot be potentially infinitely extended by using just symmetric cryptography in the presence of an adversary.

1 Introduction

When considering resource-restricted devices, public-key cryptographic protocols such as secret key exchanges and asymmetric encryption may not be practical. Without the

use of a private channel, a MRP aims at achieving data integrity with respect to the data origin. That is, the purpose of a MRP is to allow authenticated communication of messages over insecure channels. In the scenario that we have in mind, there are two honest parties, Alice and Bob, with Alice sending messages in an authenticated way to Bob, while the adversary, Eve, interferes with the communication channel.

The protocol proceeds in two phases. Initially, Alice and Bob have no common knowledge. In the *initialization* phase of the protocol, the channel between Alice and Bob is authentic, but can be eavesdropped by Eve; hence, Eve cannot alter, delete, or withhold messages during this phase. In the *main* phase of the protocol, the channel is insecure, i.e. Eve can perform active attacks. As Eve would trivially succeed, we do not consider denial-of-service attacks where Eve stops the flow of messages permanently between Alice and Bob.

We are interested in protocols that provide *authenticity* and *perenniality*. Informally, an MRP is authentic if for any message M that Bob accepts, the transmission of M was previously initiated by Alice. An MRP is perennial if all messages M, whose transmission was initiated by Alice, eventually will be accepted by Bob, provided that Eve stops active attacks at some point in the main phase. At no point in time will Alice and Bob know whether Eve has already stopped all active attacks, or will she intend to perform more of them in the future. Formal definitions of authenticity and perenniality will be provided below.

We continue the current paper with a brief survey of proposed MRPs and impossibility results in cryptography, after which we give a formalization for two-party protocols in the perfect cryptography (Dolev-Yao) model. We then give definitions of authenticity and perenniality for MRPs and show that, with the chosen set of cryptographic primitives, there can be no protocol satisfying both properties.

2 Related Work

In the literature, there have been a number of proposals for MRPs. Motivated by the use of low-cost and low-power devices such as RFID tags, Lucks et al. proposed the *Jane Doe protocol* [17] (a modified version of this protocol has been proposed by Goldberg, Mashatan, and Stinson in [19]). Earlier work includes Anderson et al.'s *Guy Fawkes protocol* [2], Mitchell's *Remote User Authentication protocol* [21], Stajano and Anderson's *Resurrecting Duckling protocol* [25], and Weimerskirch and Westhoff's *zero-common-knowledge protocol* [26]. Let's take a closer look at these protocols.

The *Jane Doe protocol* uses a hash chain to authenticate a pre-determined number of messages. For a randomly chosen $a_0 \in \{0,1\}^s$ and hash function $h : \{0,1\}^s \to \{0,1\}^s$ with key size s, the hash chain generated is $a_1 := h(a_0), \ldots, a_n := h(a_{n-1})$. Similar to the *Jane Doe protocol*, Weimerskirch and Westhoff's *zero-common-knowledge protocol* (ZCK) uses a hash chain. Unfortunately, ZCK is flawed due to Eve's ability to use a denial of service attack along with a lack of recoverability in order to convince Bob that she is Alice.

The *Guy Fawkes protocol* uses a commitment to a string that consists of the hash of a triple in the form (codeword, message, [hash of next codeword]). The first codeword needs to be bootstrapped by some external mechanism such as a conventional digital

signature or an out-of-band authentication (which may be user-aided, for example see [14]), which may be inconvenient. Furthermore, this protocol assumes that Bob can see this commitment hash before the triple is revealed by Alice. In our setting, Eve controls the flow of messages between Alice and Bob. Since the protocol does not include a provision for Alice to be sure that Bob received the commitment hash, Eve simply has to wait for Alice to reveal her codeword in order to impersonate her.

Building on the *Guy Fawkes protocol*, Mitchell's *Remote User Authentication protocol* uses a set of message authentication codes (MACs) of a random data string under different keys to authenticate a particular user (not a message). Due to the number and size of MACs used, this protocol can be expensive in terms of computation and storage. The security of the scheme depends on computational assumptions about the parameters. The number of times that this protocol can be used is limited because denial of service attacks may cause the reuse of keys during resynchronization and allow Eve to impersonate Alice.

Stajano and Anderson's *Resurrecting Duckling protocol* assumes that Alice and Bob can share an initial secret during what they refer to as the "imprinting phase". The solution proposed is physical contact between the two devices that Alice and Bob represent which may not always be feasible. As Eve is a passive observer during the *initialization* (or imprinting) phase, any information exchanged by Alice and Bob can be eavesdropped by Eve.

More recent and in the same line of research, Mashatan and Stinson's *new message recognition protocol for ad hoc pervasive networks* [18] provides a MRP of fixed size. The protocol uses a hash function to create commitment values to a current and future "password". However, as shown in [11], the resynchronization process rendered does not provide the recoverability intended, and in fact, enables an adversary to create selective forgeries.

There are not many impossibility results in cryptography, and those that exist are mostly for cryptographic primitives and certain proof methods. There are some results on the impossibility of using black-box methods for constructing one primitive from another one, e.g. collision-resistant hash functions from one-way permutations [24] or time-stamping schemes and collision-resistant hash functions from each other [7, 5, 6]. For somewhat larger systems, Backes et al. [3, 4] show that certain primitives cannot be implemented in the universally composable cryptographic library in a certain reasonable way.

Regarding protocols, there is a well-known result stating that a fair exchange protocol cannot be built without a trusted third party [27]. Impagliazzo's and Rudich's result [12] on the impossibility of establishment of a common secret over an authentic channel is maybe the closest to what we achieve in the current paper, but in some sense, it is the weakest of the ones listed here — it shows that if one manages to prove that secret agreement is possible assuming only that one-way permutations exist, then one has proved $\mathbf{P} \neq \mathbf{NP}$. Another result on the non-existence of a certain class of protocols is by Pereira and Quisquater [22] which shows that Diffie-Hellman based group key exchange protocols cannot be constructed if the parties are constrained to perform only exponentiations in the underlying group, and only elements of the group may be exchanged between parties.

3 Protocols and Execution

3.1 Messages

To be able to show the non-existence of a certain class of protocols, we have to specify what a protocol is. We are working in the *perfect cryptography* (or *Dolev-Yao*) model [9]. Messages are modeled as elements of a term algebra, the operations possible with the messages are explicitly listed, and the adversary is bound to the same list.

Let $\mathbf{R}$ be a countable set of *formal nonces*, $\mathbf{C}$ a countable set of *formal constants*, and $\mathbf{P}$ a countable set of *formal payloads*. Let the sets $\mathbf{R}$, $\mathbf{C}$ and $\mathbf{P}$ be mutually disjoint. Let $\mathbf{A} = \mathbf{R} \cup \mathbf{C} \cup \mathbf{P}$. The set of formal *pre-messages* $\Sigma^\#$ is defined as the smallest set satisfying

- $\mathbf{A} \subseteq \Sigma^\#$;
- if $m_1, \ldots, m_k \in \Sigma^\#$ then $h(m_1, \ldots, m_k) \in \Sigma^\#$;
- if $m_1, m_2 \in \Sigma^\#$ then $(m_1 \oplus m_2) \in \Sigma^\#$.

We say that $h(m_1, \ldots, m_k)$ is the *formal hash* of messages $m_1, \ldots, m_k$ and $(m_1 \oplus m_2)$ is the *exclusive or* (XOR) of the messages m_1, m_2.

The set of *formal messages* Σ is defined as the factor set $\Sigma^\#/\equiv$ where $\equiv$ relates two messages that we want to consider equal because of the properties of $\oplus$. We assume there is a fixed element $0 \in \mathbf{C}$. The relation $\equiv$ is the least congruence (with respect to the operations h and $\oplus$) that contains $x \oplus y \equiv y \oplus x$, $x \oplus x \equiv 0$, $0 \oplus x \equiv x$ and $(x \oplus y) \oplus z \equiv x \oplus (y \oplus z)$ for all $x, y, z \in \Sigma^\#$. In the set Σ, we consider $\oplus$ to be a long operation, taking any number of arguments, because of the associativity imposed by $\equiv$.

We define the relation "is submessage of" (denoted $\sqsubseteq$) on messages. We define $m \sqsubseteq m$ for all messages m, and if $m' \sqsubseteq m$, then also $m' \sqsubseteq h(\ldots, m, \ldots)$ and $m' \sqsubseteq (\ldots \oplus m \oplus \ldots)$.

Given a set of messages $\mathbf{M}$, we can construct new messages m from it (denoted $\mathbf{M} \vdash m$) only in the following ways:

1. $\mathbf{M} \vdash m$ for any $m \in \mathbf{M}$;
2. $\mathbf{M} \vdash C$ for any $C \in \mathbf{C}$;
3. if $\mathbf{M} \vdash m_1, \ldots, \mathbf{M} \vdash m_k$ then $\mathbf{M} \vdash h(m_1, \ldots, m_k)$;
4. if $\mathbf{M} \vdash m_1, \ldots, \mathbf{M} \vdash m_k$ then $\mathbf{M} \vdash (m_1 \oplus \cdots \oplus m_k)$.

Denote $\langle \mathbf{M} \rangle \stackrel{\triangle}{=} \{m \in \Sigma \,|\, \mathbf{M} \vdash m\}$.

In addition, we want to consider more limited constructions. We denote by $\mathbf{M} \vdash_h m$ the construction of new messages m from $\mathbf{M}$ using properties (1), (2), and (3) from above. Likewise, $\mathbf{M} \vdash_\oplus m$ denotes the construction of new messages m from $\mathbf{M}$ using properties (1), (2), and (4). Also, two messages can be compared for equality, and given a message, it is possible to check whether it is a constant or a payload. Note that we assume that a nonce cannot be told apart from a formal hash. Indeed, they both model "random-looking" bitstrings.

3.1.1 Modeling Symmetric Cryptography

When using perfect cryptography to model protocols, one usually understands certain sets of cryptographic primitives under the notions of "symmetric cryptography" or "asymmetric cryptography". Symmetric cryptography usually includes not only symmetric encryption and hash functions, but also message authentication codes, (pseudo)random functions, and permutations. It may also include XOR or other computationally simple operations with data. On the other hand, asymmetric cryptography contains primitives like public-key encryption and signing using operations like exponentiation (to model Diffie-Hellman key exchange).

In the current paper, we explicitly consider only hash functions and the XOR operation. Nevertheless, we claim that we are still handling most of "symmetric cryptography" because other primitives under this label can be constructed from hashes and XORs in a manner that a protocol using the atomic primitive is indistinguishable from a protocol using the constructed primitive [13]. For example, (randomized) symmetric encryption can be defined as $\mathcal{E}^r_K(m) = (r, h(K, h(K, r, m)) \oplus m, h(K, r, m))$. A pseudorandom function can be defined as $PRF_K(m) = h(K, m)$. A message authentication code can be defined exactly in the same way. A pseudorandom permutation can be constructed from a pseudorandom function by using the Feistel construction [10, Sec. 3.7.2].

3.2 Alice and Bob

The MRP proceeds in rounds, i.e. we assume a global clock. The construction of protocols is generally easier in the synchronous model, hence this assumption strengthens our impossibility result. During a round, Alice and Bob read the messages sent to them during the previous round (possibly modified by Eve), generate new messages, and send them to each other (possibly captured by Eve). Additionally, at the beginning of a round, Alice may receive a *payload* that she must somehow transmit to Bob. Also, in addition to sending messages, Bob may also choose to *accept* payloads.

If Eve is not active then the messages sent by Alice and Bob are handed to each other at the beginning of the next round. Otherwise, Eve receives those messages and replaces them with messages of her choosing.

Recall that the protocol had two phases. During the initialization phase, Eve is not active. The end of the initialization phase is denoted by Alice (this is w.l.o.g. as Alice and Bob can discuss when to start with the main phase). During the main phase, Eve becomes active. She starts interfering with the communication between Alice and Bob. At some point, Eve may decide to become inactive again. When this happens, Alice and Bob will get no notification.

Formally, the protocol role for Alice is defined by the following components:

- The set of *internal states* $\mathbf{S}_A$ (possibly infinite) and the initial state $S_{A0} \in \mathbf{S}_A$ of Alice.
- The *transition function* δ_A whose type is described below.

The arguments to the transition function are the following:

- the current state $S^\circ_A \in \mathbf{S}_A$;

- the current *message store*, $\mathcal{M}^\circ \in \Sigma^*$ (where X^* denotes the set of finite sequences of elements of the set X);
- the sequence of messages $\mathcal{M}_{A \leftarrow B} \in \Sigma^*$ received at the beginning of the current round, presumably from Bob;
- the payloads $\mathcal{M}_{\mathrm{pl}} \in \mathbf{P}^*$ that Alice received from Eve to be transmitted to Bob (possibly empty).

Alice's transition function outputs the following components:

- new internal state $S_A^\bullet \in \mathbf{S}_A$;
- new store of messages $\mathcal{M}^\bullet \in \Sigma^*$;
  - We demand that $\mathcal{M}^\bullet = \mathcal{M}^\circ \cdot \mathcal{M}_{\mathrm{pl}} \cdot \mathcal{M}_{A \leftarrow B} \cdot \mathcal{N}$, where $\mathcal{N} \in \mathbf{R}$ is the sequence of formal nonces generated by Alice in the current round.
- the sequence of messages $\mathcal{M}_{A \rightarrow B} \in \Sigma^*$ to be sent to Bob;
- a Boolean b_{m} indicating whether the main phase of the protocol should start (this component is ignored after the main phase has started).

The transition function must satisfy certain properties, described below. These properties state that Alice can construct and compare the formal messages only according to the rules given above.

Similarly, Bob's role is defined by its set of internal states $\mathbf{S}_B$, the initial state $S_{B0} \in \mathbf{S}_B$ and the transition function δ_B. The inputs and outputs of δ_B are the same as of δ_A, except

- there is no input $\mathcal{M}_{\mathrm{pl}}$ nor output b_{m};
- there is an additional output $\mathcal{M}_{\mathrm{acc}} \in \mathbf{P}^*$ of payloads Bob has accepted during the current round.

The conditions on δ_A and δ_B are inspired by the formal meaning of epistemic modalities in authentication logics [1]. Similarly to those models, Alice and Bob can only act upon the information that they actually have. Given two message stores that look the same to Alice, her outputs cannot allow her to distinguish these two stores. Let $\mathcal{M}$ and $\mathcal{M}'$ be two message stores of the same length ℓ. Let $\mathcal{M}[i]$ denote the message at the i-th position of $\mathcal{M}$, where $1 \leq i \leq \ell$. We say that $\mathcal{M}$ and $\mathcal{M}'$ are *indistinguishable* (denote $\mathcal{M} \approx \mathcal{M}'$) if there exists an *isomorphism* φ from $\langle \mathcal{M} \rangle$ to $\langle \mathcal{M}' \rangle$, such that $\varphi(\mathcal{M}[i]) = \mathcal{M}'[i]$ for all $i \in \{1, \ldots, \ell\}$. A mapping φ from a set of messages X to a set Y is an *isomorphism* if it is bijective and

- $\forall c \in \mathbf{C} : \ \varphi(c) = c$;
- $\forall m \in X$: m is a payload iff $\varphi(m)$ is a payload;
- $\forall m_1, \ldots, m_k \in X : \varphi(h(m_1, \ldots, m_k)) = h(\varphi(m_1), \ldots, \varphi(m_k))$ and $\varphi(m_1 \oplus \cdots \oplus m_k) = \varphi(m_1) \oplus \cdots \oplus \varphi(m_k)$.

Let $S^\circ \in \mathbf{S}_A$ and let $\mathcal{M}^\circ, \mathcal{M}^{\circ\prime}, \mathcal{M}_{A\leftarrow B}, \mathcal{M}'_{A\leftarrow B}, \mathcal{M}_{\mathrm{pl}}, \mathcal{M}'_{\mathrm{pl}}$ be sequences of messages. Let

$$(S^\bullet, \mathcal{M}^\bullet, \mathcal{M}_{A\to B}, b) = \delta_A(S^\circ, \mathcal{M}^\circ, \mathcal{M}_{A\leftarrow B}, \mathcal{M}_{\mathrm{pl}})$$
$$(S^{\bullet\prime}, \mathcal{M}^{\bullet\prime}, \mathcal{M}'_{A\to B}, b') = \delta_A(S^\circ, \mathcal{M}^{\circ\prime}, \mathcal{M}'_{A\leftarrow B}, \mathcal{M}'_{\mathrm{pl}}) \ .$$

If

- $|\mathcal{M}^\circ| = |\mathcal{M}^{\circ\prime}|$, $|\mathcal{M}_{A\leftarrow B}| = |\mathcal{M}'_{A\leftarrow B}|$, $|\mathcal{M}_{\mathrm{pl}}| = |\mathcal{M}'_{\mathrm{pl}}|$,
- $\mathcal{M}^\circ \cdot \mathcal{M}_{\mathrm{pl}} \cdot \mathcal{M}_{A\leftarrow B} \approx \mathcal{M}^{\circ\prime} \cdot \mathcal{M}'_{\mathrm{pl}} \cdot \mathcal{M}'_{A\leftarrow B}$, where the indistinguishability is realized by the isomorphism φ°

then the following must hold:

- $S^\bullet = S^{\bullet\prime}$; $b = b'$; $|\mathcal{M}^\bullet| = |\mathcal{M}^{\bullet\prime}|$; $|\mathcal{M}_{A\to B}| = |\mathcal{M}'_{A\to B}|$;
- $\mathcal{M}^\bullet \cdot \mathcal{M}_{A\to B} \approx \mathcal{M}^{\bullet\prime} \cdot \mathcal{M}'_{A\to B}$ where the indistinguishability can be realized by some isomorphism $\varphi^\bullet$ that extends φ°.

Similar condition (indistinguishable inputs lead to indistinguishable outputs) must hold for δ_B. The isomorphism on inputs obviously does not include $\mathcal{M}_{\mathrm{pl}}$, but the isomorphism on outputs also has to include $\mathcal{M}_{\mathrm{acc}}$.

3.3 Global Setup

The *global state* $\mathcal{S}$ of the protocol (between the rounds) consists of the following parts:

- the states S_A, S_B and message stores $\mathcal{M}_A$, $\mathcal{M}_B$ of Alice and Bob;
  - initially S_{A0}, S_{B0}, and empty
- the set of messages $\mathbf{M}_E$ that Eve has seen or generated;
  - initially $\langle\emptyset\rangle$
- the Booleans b_{m}, b_{a} indicating whether the protocol execution is in the main phase, and whether Eve is active;
  - initially false and true, respectively
- the sequences of messages $\bar{\mathcal{M}}_{A\leftarrow B}$ and $\bar{\mathcal{M}}_{B\leftarrow A}$ that Alice and Bob are about to receive;
  - initially empty
- the sequence $\mathcal{M}_{\mathrm{pl}}$ of payloads that Alice should transmit to Bob.
  - initially empty

We say that global state $\mathcal{S}$ is transformed to $\mathcal{S}'$ in a single round and write $\mathcal{S} \to \mathcal{S}'$ if the following holds. Let

$$\begin{aligned}(S'_A, \mathcal{M}'_A, \mathcal{M}_{A\to B}, b) &= \delta_A(S_A, \mathcal{M}_A, \bar{\mathcal{M}}_{A\leftarrow B}, \mathcal{M}_{\mathrm{pl}})\\ (S'_B, \mathcal{M}'_B, \mathcal{M}_{B\to A}, \mathcal{M}_{\mathrm{acc}}) &= \delta_B(S_B, \mathcal{M}_B, \bar{\mathcal{M}}_{B\leftarrow A})\end{aligned} \tag{1}$$

Then $S'_A, S'_B, \mathcal{M}'_A, \mathcal{M}'_B$ must be components of $\mathcal{S}'$. The other components of $\mathcal{S}'$ must satisfy the following conditions.

- If $b_{\mathrm{m}} \wedge b_{\mathrm{a}}$ then there must exist a finite set of nonces and payloads $\mathbf{N} \subset \mathbf{R} \cup \mathbf{P}$ not occurring in $\mathcal{S}$, $\mathcal{M}'_A$ and $\mathcal{M}'_B$, such that $\mathbf{M}'_E = \langle \mathbf{M}_E \cup \mathcal{M}_{A\to B} \cup \mathcal{M}_{B\to A} \cup \mathbf{N} \rangle$. Otherwise $\mathbf{M}'_E = \langle \mathbf{M}_E \cup \mathcal{M}_{A\to B} \cup \mathcal{M}_{B\to A} \rangle$.
- $b'_{\mathrm{m}} = b_{\mathrm{m}} \vee b$.
- If b_{a} is false then b'_{a} must be false.
- If $b_{\mathrm{m}} \wedge b_{\mathrm{a}}$ then the components of the sequences of messages $\bar{\mathcal{M}}'_{A\leftarrow B}$ and $\bar{\mathcal{M}}'_{B\leftarrow A}$ belong to $\mathbf{M}'_E$. Otherwise $\bar{\mathcal{M}}'_{A\leftarrow B} = \mathcal{M}_{B\to A}$ and $\bar{\mathcal{M}}'_{B\leftarrow A} = \mathcal{M}_{A\to B}$.
- If $b_{\mathrm{m}} \wedge b_{\mathrm{a}}$ then $\mathcal{M}'_{\mathrm{pl}}$ is a possibly empty sequence of payloads that belong to $\mathbf{M}'_E$. Otherwise $\mathcal{M}'_{\mathrm{pl}}$ is empty.

We see that Eve acts only if both flags b_{m} and b_{a} are set. In this case she non-deterministically selects the messages received by A and B. If Eve does not act then $\mathcal{S}'$ is uniquely determined by $\mathcal{S}$.

A *protocol trace* is an infinite sequence $\mathcal{S}_0 \to \mathcal{S}_1 \to \cdots$, such that $\mathcal{S}_0$ is the initial global state described above and for each i we have $\mathcal{S}_{i-1} \to \mathcal{S}_i$.

3.4 Security Properties

We say that Bob *accepts* payload M at the step $\mathcal{S} \to \mathcal{S}'$ if in the equation (1), the component $\mathcal{M}_{\mathrm{acc}}$ contains M. We say that Alice *initiates* the payload M in the state $\mathcal{S}$, if the component $\mathcal{M}_{\mathrm{pl}}$ of that state contains M.

We say that the MRP is **authentic** if the following holds for all of its traces $\mathcal{S}_0 \to \mathcal{S}_1 \to \cdots$. If Bob accepts a payload M at the step $\mathcal{S}_i \to \mathcal{S}_{i+1}$ then there exists $j \in \{1, \ldots, i\}$, such that Alice initiates M in the state $\mathcal{S}_j$.

We say that the MRP is **perennial** if the following holds for all of its traces $\mathcal{S}_0 \to \mathcal{S}_1 \to \cdots$. If Alice initiates the payload M in some state $\mathcal{S}_i$, and there exists a state $\mathcal{S}_k$, where b_{a} is false, then there exists some j, such that Bob accepts M at the step $\mathcal{S}_j \to \mathcal{S}_{j+1}$.

Theorem 1. *There exist no authentic perennial MRPs using hashing as the only cryptographic primitive.*

4 Proof of the Theorem

We give a constructive proof, describing how Eve should attack the protocol. We explain how Eve must construct the messages received by Alice and Bob. According to the previous section, Eve is actually non-deterministic, hence our descriptions serve to point out a trace where either the authenticity or perenniality is violated.

During the protocol run, Alice, Bob and Eve construct new messages from messages already in their message stores. We now formally define the procedure they use to do it. Let $\mathcal{V} = \{v_1, \ldots, v_k\}$ be a set of variables. The set $\mathcal{P}$ of *message contexts* over $\mathcal{V}$ is defined as the smallest set satisfying

- $\mathcal{V} \cup \mathbf{C} \subseteq \mathcal{P}$,
- if $P_1, \ldots, P_k \in \mathcal{P}$ then also $h(P_1, \ldots, P_k) \in \mathcal{P}$ and $P_1 \oplus \cdots \oplus P_k \in \mathcal{P}$.

Given messages $m_1, \ldots, m_k$, the notation $P[m_1, \ldots, m_k]$ denotes the message one obtains from P by substituting v_i with m_i.

We begin our proof in Sec. 4.1 by showing that Alice and Bob cannot perform a secret key exchange using formal hashing and/or $\oplus$. We then proceed to show that during the main phase of the protocol, Eve can force Alice to deplete the means with which she can prove the authenticity of her messages to Bob. In order to not overwhelm the reader with details, in Sec. 4.2, we first give the proof under the assumption that Alice and Bob do not use the $\oplus$-operation in their computations. In this case, obviously, Eve has no use of this operation as well. After that, in Sec. 4.3, we show how the proof carries over to the full language of messages.

4.1 Impossibility of Secret Key Exchange

The impossibility of the derivation of a common secret by Alice and Bob is a direct consequence of Schmidt et al. [23]. We provide the proof here for completeness.

Let $\mathcal{S}_i$ be a global state reached during the initialization phase of the protocol. Suppose that the derivation of a common secret (a message known by Alice and Bob, but not Eve) was impossible from the message stores of Alice and Bob in state $\mathcal{S}_{i-1}$. We show that it is impossible in state $\mathcal{S}_i$ as well.

Alice's message store contains the nonces that she has generated and messages that she has received from Bob (later, it also contains payloads received from Eve). Bob's message store contains nonces that he has generated and messages he has received from Alice. All exchanged messages are known to Eve. Alice now performs computations from the messages her store contains, and Bob also performs computations from his store. We describe an intermediate stage of this computation by stating that three sets of messages, $\mathbf{M}_A$, $\mathbf{M}_B$ and $\mathbf{M}_E$ are a *partial computation* if

- $\mathbf{M}_A \cap \mathbf{M}_B \subseteq \langle \mathbf{M}_E \rangle$.
- If $m \in \mathbf{M}_A \cup \mathbf{M}_B$ and $m' \sqsubseteq m$, then $m' \in \mathbf{M}_A \cup \mathbf{M}_B$.
- For any message $m = h(m_1, \ldots, m_k)$ in $\mathbf{M}_A$ [resp. $\mathbf{M}_B$]: if some of the messages $m_1, \ldots, m_k$ do not belong to $\mathbf{M}_A$ [resp. $\mathbf{M}_B$] then $m \in \mathbf{M}_B$ [resp. $m \in \mathbf{M}_A$].

Alice's and Bob's message stores $\mathcal{M}_A$ and $\mathcal{M}_B$ together with Eve's knowledge $\mathbf{M}_E$ in $\mathcal{S}_i$ do not form a partial computation, because the stores only contain nonces and received messages but not intermediate computations. But if we define $\mathbf{M}_A$ as $\mathcal{M}_A$ together with the results of all computation steps Alice has performed so far during the execution of the protocol in order to construct the messages for Bob, and define $\mathbf{M}_B$ similarly from $\mathcal{M}_A$, then we end up with a partial computation. Indeed, the submessages of all messages in $\mathcal{M}_A \cup \mathcal{M}_B$ must be in this set too. Also, each message of the form $h(m_1, \ldots, m_k)$ had to be computed by either Alice or Bob, and the one who computed it must have known the components $m_1, \ldots, m_k$. Finally, any message in both $\mathbf{M}_A$ and $\mathbf{M}_B$ is also in $\mathbf{M}_E$ (i.e. there are no common secrets). Let $\mathcal{M}_A^{\text{prev}}$ [resp. $\mathcal{M}_B^{\text{prev}}$] be the message store of Alice [resp. Bob] in the state $\mathcal{S}_{i-1}$. No common secrets were derivable in $\mathcal{S}_{i-1}$, i.e. all messages in $\langle \mathcal{M}_A^{\text{prev}} \rangle \cap \langle \mathcal{M}_B^{\text{prev}} \rangle$ were known to Eve. But each of the messages in $\mathbf{M}_A$ [resp. $\mathbf{M}_B$] either belongs to $\langle \mathcal{M}_A^{\text{prev}} \rangle$ [resp. $\mathcal{M}_B^{\text{prev}}$] or was received from Bob [resp. Alice] during the step $\mathcal{S}_{i-1} \to \mathcal{S}_i$ and is thus known to Eve.

We now show that neither XOR nor hashing help Alice and Bob derive a common secret from a partial computation.

Lemma 2. *Let $\mathbf{M}_A$, $\mathbf{M}_B$, $\mathbf{M}_E$ be a partial computation. Let $X_A \subseteq \mathbf{M}_A$, $X_B \subseteq \mathbf{M}_B$ be finite sets. Let $s_A = \bigoplus X_A$ and $s_B = \bigoplus X_B$. If $s_A = s_B = s$, then $\mathbf{M}_E \vdash s$.*

Proof. Consider the sets X_A and X_B, and suppose that $s_A = s_B$. First, we show that we can assume that all messages in X_A and X_B are nonces or formal hashes. Indeed, if a message $m = m_1 \oplus \cdots \oplus m_k$ belongs to (say) X_A, then we could replace it with the messages $m_1, \ldots, m_k$ which must exist in either $\mathbf{M}_A$ or $\mathbf{M}_B$ by the definition of partial computation. Each of the messages m_i will be added to either X_A or X_B (or removed, if it already exists there), depending on whether $\mathbf{M}_A$ or $\mathbf{M}_B$ contains it.

If only nonces and formal hashes are elements of X_A and X_B, then nothing "cancels out" when we compute $s_A = \bigoplus X_A$ and $s_B = \bigoplus X_B$. Hence, if $s_A = s_B = s$, then $X_A = X_B = X$ and $X \subseteq \mathbf{M}_A \cap \mathbf{M}_B$. Therefore, Eve knows all elements of X and can compute s herself. □

Lemma 3. *Let $\mathbf{M}_A$, $\mathbf{M}_B$, $\mathbf{M}_E$ be a partial computation. Let $m_1, \ldots, m_k$ in $\mathbf{M}_A$. Let $m = h(m_1, \ldots, m_k)$. If $m \in \mathbf{M}_B$, then $\mathbf{M}_E \vdash m$. The same result holds if we swap $\mathbf{M}_A$ and $\mathbf{M}_B$.*

Proof. If $m \in \mathbf{M}_B$, then either $m_1, \ldots, m_k \in \mathbf{M}_B$ or $m \in \mathbf{M}_A$ by the definition of a partial computation. In the second case, $m \in \mathbf{M}_A \cap \mathbf{M}_B$, thus Eve knows m. In the first case, the premise of the lemma stating that $m_1, \ldots, m_k \in \mathbf{M}_A$ implies that Eve already knows $m_1, \ldots, m_k$ and can thus compute m herself. □

Lemma 4. *Let $\mathbf{M}_A$, $\mathbf{M}_B$, $\mathbf{M}_E$ be a partial computation. Let $m_1, \ldots, m_k \in \mathbf{M}_A$ and let m be computed as $m = h(m_1, \ldots, m_k)$ or $m = m_1 \oplus \cdots \oplus m_k$. Then $\mathbf{M}_A \cup \{m\}$, $\mathbf{M}_B$, $\mathbf{M}_E$ is a partial computation too. The same result holds if we swap $\mathbf{M}_A$ and $\mathbf{M}_B$.*

Proof. The structural properties of messages in $\mathbf{M}_A \cup \{m\}$ and $\mathbf{M}_B$ are obviously satisfied — the immediate submessages of m are already elements of $\mathbf{M}_A$ (or in case of XOR, possibly $\mathbf{M}_B$). Also, if $m \in \mathbf{M}_B$, then the two previous lemmas imply that $m \in \langle \mathbf{M}_E \rangle$ as well. □

The last lemma shows that if Alice and Bob perform computations with the values that they know after a step in the initialization phase of the protocol, and they had no common secrets before that step, then the messages they know can only be partial computations. As a result, they can derive no common secrets. The presented lemmas also show that no common secret can be obtained during the main phase of the protocol — although $\mathbf{M}_A$, $\mathbf{M}_B$ and $\mathbf{M}_E$, where $\mathbf{M}_A$ and $\mathbf{M}_B$ are defined as $\mathcal{M}_A$ and $\mathcal{M}_B$ together with the results of all intermediate computations of Alice and Bob do not yet form a partial computation (because a message learned by Alice may be originated by Eve and not by Bob), the sets $\mathbf{M}_A \cup \langle \mathbf{M}_E \rangle$, $\mathbf{M}_B \cup \langle \mathbf{M}_E \rangle$ and $\mathbf{M}_E$ form a partial computation as long as there were no common secrets before the current round. Hence, Alice and Bob cannot derive a common secret even if given access to everything that Eve knows.

4.2 Proof for Language without XOR

Suppose that the initialization phase of the protocol has just ended — at the step $\mathcal{S}_{i-1} \to \mathcal{S}_i$, Alice decided that the main phase should start.

Let $\mathbf{Z}^0$ be the set of all messages that Alice and Bob have sent during the initialization phase. Consider the knowledge $\mathbf{M}_E$ of Eve in the state $\mathcal{S}_i$. Let $\mathbf{Z} = \{m' \,|\, m \in \mathbf{Z}^0, m' \sqsubseteq m\} \backslash \mathbf{M}_E$, i.e. $\mathbf{Z}$ contains all submessages of sent messages that Eve does not know. Because of the results of Sec. 4.1, each element of $\mathbf{Z}$ is known to exactly one of Alice and Bob. Let $\mathbf{Z} = \mathbf{Z}_A \dot\cup \mathbf{Z}_B$, where $\mathbf{Z}_A$ [resp. $\mathbf{Z}_B$] is the set of messages in $\mathbf{Z}$ known only to Alice [resp. Bob].

W.l.o.g., we partition the set of formal nonces $\mathbf{R}$ into three countable sets $\mathbf{R}_A$, $\mathbf{R}_B$ and $\mathbf{R}_E$ and assume that whenever Alice, Bob, or Eve generates a new nonce, it comes from the respective set. Let $\mathbf{Y}_A = \mathbf{Z}_A \cup \mathbf{R}_A$ and $\mathbf{Y}_B = \mathbf{Z}_B \cup \mathbf{R}_B$. We now define mappings tr_A and tr_B from messages to messages as follows:

$$tr_A(m) = \begin{cases} m, & m \in \mathbf{Z}^0 \cup \mathbf{C} \\ \square^m, & m \in \mathbf{Y}_A \\ m', & m = \square^{m'}, m' \in \mathbf{Y}_B \\ h(tr_A(m_1), \ldots, tr_A(m_k)), & m = h(m_1, \ldots, m_k) \end{cases}$$

$$tr_B(m) = \begin{cases} m, & m \in \mathbf{Z}^0 \cup \mathbf{C} \\ \square^m, & m \in \mathbf{Y}_B \\ m', & m = \square^{m'}, m' \in \mathbf{Y}_A \\ h(tr_B(m_1), \ldots, tr_B(m_k)), & m = h(m_1, \ldots, m_k), \end{cases}$$

where the different cases have to be considered from top to bottom. Here $\square^m \in \mathbf{R}_E$ is a new nonce that Eve constructs the first time that she needs to consider the second case for the message m. Additionally, we state that tr_A is a permutation on payloads (but do not specify which one). The mapping tr_B is also a permutation on payloads and it is equal to the inverse of tr_A.

In the main phase of the protocol run, as long as Alice and Bob do not send each other the messages in $\mathbf{Z}$, the attack mounted by Eve consists of replacing all messages m sent by Alice with $tr_A(m)$, and all messages m sent by Bob with $tr_B(m)$. We explain below what happens if some message from the set $\mathbf{Z}$ is sent.

Lemma 5. *Eve is capable of replacing all messages $m \notin \mathbf{Z}$ sent by Alice with $tr_A(m)$, and all messages sent by Bob with $tr_B(m)$.*

Proof. Let m be a message sent by Alice. Alice constructed it as $P[\vec{z}, \vec{y}, \vec{x}]$ for a certain message context P and messages $\vec{z} \in \mathbf{Z}^0 \cup \mathbf{C} \cup \mathbf{P}$ (this notation means that each element of the vector $\vec{z}$ belongs to the set $\mathbf{Z}^0 \cup \mathbf{C} \cup \mathbf{P}$), messages $\vec{y} \in \mathbf{Y}_A$, and messages $\vec{x}$ presumably received from Bob after the start of the main phase. Those messages $\vec{x}$ were actually generated by Eve from the messages $\overrightarrow{tr_B^{-1}(x)}$ actually constructed by Bob. Eve can now construct $tr_A(m)$ as $P[\vec{z}, \overrightarrow{\square^y}, \overrightarrow{tr_B^{-1}(x)}]$. The message m sent by Bob is translated to $tr_B(m)$ in the same way. □

Let us now show that Alice and Bob do not notice Eve replacing the exchanged messages m with $tr_A(m)$ or $tr_B(m)$. Let $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ be two global states. Let $\mathbf{Z}^0$ be a set of messages that Alice, Bob, and Eve all know in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$. Also, let each of the messages in $\mathbf{Z}^0$ appear in the message store of either Alice or Bob as a message from the other party. Let $\mathbf{Z}$ be the set of submessages of $\mathbf{Z}^0$ unknown to Eve and known to exactly one of Alice and Bob. The sets $\mathbf{Z}^0$ and $\mathbf{Z}$ must look like the sets of messages and their submessages of an initial segment of a conversation between Alice and Bob. That is, there must exist an order on $\mathbf{Z}^0$ such that each message in $\mathbf{Z}^0$ can be constructed from previous messages of $\mathbf{Z}^0$, from the nonces in $\mathbf{Z}$, and from the nonces in $\mathbf{R}_E$.

Define tr_A and tr_B as above. Let the states be *isomorphic* (denoted $\mathcal{S} \cong \mathcal{S}'$), meaning that

- The internal states of Alice and Bob are the same in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$;
- $\mathcal{M}^\circ \cdot \mathcal{M}_{\text{pl}} \cdot \mathcal{M}_{A \leftarrow B} \approx \mathcal{M}^{\circ\prime} \cdot \mathcal{M}'_{\text{pl}} \cdot \mathcal{M}'_{A \leftarrow B}$, where the isomorphism φ_A is the following:
  - $\varphi_A(x) = x$ if $x \in \mathbf{R}_A$ or $x \in \mathbf{P}$,
  - $\varphi_A(m) = tr_B(m)$ if m is a message received from Bob
    - in particular, $\varphi_A(m) = m$ for all $m \in \mathbf{Z}^0$

  (recall that message stores of Alice consist of nonces generated by her, payloads, and messages received from the network);
- the message stores of Bob must be isomorphic too, where the isomorphism φ_B is identity on nonces Bob has generated, and equals tr_A on messages received from Alice.

Let $\mathcal{S}^\circ \to \mathcal{S}^\bullet$, where the step corresponds to Eve not interfering with the messages Alice and Bob are sending to each other. Also let $\mathcal{S}^{\circ\prime} \to \mathcal{S}^{\bullet\prime}$ where the step corresponds to Eve applying tr_A to the messages Alice is sending, and tr_B to the messages Bob is sending, before forwarding them to the other party.

Lemma 6. *Let the sets $\mathbf{Z}$ and $\mathbf{Z}^0$ be as defined above. Let $\mathcal{M}$ be a message store of Alice. Let φ be a mapping from $\mathcal{M}$ to the set of all messages Σ defined as follows: $\varphi(x) = x$ if $x \in \mathbf{R}_A$ or $x \in \mathbf{P}$, and $\varphi(x) = tr_B(x)$ otherwise. Let y be a message that satisfies the following:*

- $y \notin \mathbf{Z}$.

- *If $r \in \mathbf{Y}_A$ and $r \sqsubseteq y$, then exists $z \in \mathbf{Z}^0$, such that $r \sqsubseteq z \sqsubseteq y$. Such z exists for each occurrence of r in y.*

Let $P[\vec{x}] = y$ for a certain message context P and messages $\vec{x}$ in $\mathcal{M}$. Then $P[\overrightarrow{\varphi(x)}] = tr_B(y)$.

Proof. Induction over the length of computing $tr_B(y)$. If $y \in \mathbf{Z}^0 \cup \mathbf{C}$ then $tr_B(y) = y$. As $P[\vec{x}] = y$, all members of the vector $\vec{x}$ must be submessages of y. Hence they're elements of $\mathbf{Z}^0$, $\mathbf{Z}$ or $\mathbf{P}$. All elements in Alice's message store that are also elements of $\mathbf{Z}$ are nonces generated by Alice. Hence $\varphi(x) = x$ for all elements x of the vector $\vec{x}$ and $P[\overrightarrow{\varphi(x)}] = P[\vec{x}] = y = tr_B(y)$.

If $y \in \mathbf{Y}_B$ then $tr_B(y) = \square^y$. But in this case $P[\vec{x}] = y$ is impossible because only Bob knows elements in $\mathbf{Y}_B$. Also, the case $y = \square^{m'}$, where $m' \in \mathbf{Y}_A$ is impossible too, because Alice does not generate nor receive nonces of the form $\square^r$ where r is a secret known to Alice.

If $\mathbf{Z}^0 \not\ni y = h(y_1, \ldots, y_k)$ then we consider the shape of the context P. If P is a single variable x_i, then the message corresponding to x_i cannot be a nonce or payload, hence $\varphi(x_i) = tr_B(x_i)$. Also, the message corresponding to the variable x_i is equal to y. We have $\varphi(x_i) = tr_B(x_i) = tr_B(y)$. If $P = h(P_1, \ldots, P_k)$ then $P_i[\vec{x}] = y_i$ for all $i \in \{1, \ldots, k\}$. The computation of $tr_B(y_i)$ requires fewer steps than the computation of $tr_B(y)$. Hence, if $y_i \notin \mathbf{Z} \cup \mathbf{R}_A$, then we can apply the induction assumption to $P_i[\vec{x}]$ and y_i and obtain $P_i[\overrightarrow{\varphi(x)}] = tr_B(y_i)$. The case $y_i \in \mathbf{Y}_A$ is impossible. If $y_i \in \mathbf{Y}_B$, then Alice would again be capable of computing an element of $\mathbf{Y}_B$ as $P_i[\vec{x}]$, hence this case is impossible too. □

Lemma 7. *If the messages sent by Alice and Bob in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ do not contain elements of $\mathbf{Z}$, then in states $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$, the messages in the set $\mathbf{Z}$ are still known by exactly one of Alice and Bob. Furthermore, $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$ are isomorphic, and the isomorphism between messages is related to tr_A and tr_B in the same way as for $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$.*

Proof. First, we note that because of the conditions put on δ_A and δ_B (Alice and Bob cannot act on information they do not have), Alice's and Bob's internal states are the same in $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$. Also, the message tuples $\mathcal{M}_{A\to B}$ and $\mathcal{M}'_{A\to B}$ that Alice sends to Bob in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ are isomorphic; the isomorphism is given by φ that has been extended to the new nonces that Alice generated. The same holds for the tuples $\mathcal{M}_{B\to A}$ and $\mathcal{M}'_{B\to A}$ sent by Bob.

As Alice and Bob are not sending any elements of $\mathbf{Z}$ to each other, they cannot learn any of its elements. The proof of Alice and Bob not learning any messages in $\mathbf{Z}$ is given in Sec 4.1.

We now turn our attention to the isomorphism of Alice's views in $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$. A mapping φ_A is defined on the components $\mathcal{M}_A \cdot \mathcal{M}_{\text{pl}} \cdot \mathcal{M}_{A\leftarrow B}$ of $\mathcal{S}^\bullet$ by $\varphi_A(x) = x$ for $x \in \mathbf{R}_A$ and $\varphi_A(m) = tr_B(m)$ for other messages (by induction assumption on $\mathcal{M}_A$ and by Eve's behavior on $\mathcal{M}_{\text{pl}}$ and $\mathcal{M}_{A\leftarrow B}$). We show that φ_A can be extended to a homomorphism of messages.

The inability to extend φ_A to a homomorphism means that there are message contexts P, Q and elements $\vec{x}, \vec{y} \in \mathcal{M}_A \cup \mathcal{M}_{\text{pl}} \cup \mathcal{M}_{A\leftarrow B}$, such that $P[\vec{x}] = Q[\vec{y}]$, but $P[\overrightarrow{\varphi_A(x)}] \neq Q[\overrightarrow{\varphi_A(y)}]$. If the topmost operation of both P and Q is hashing, then $P = h(P_1, \ldots, P_k)$, $Q = h(Q_1, \ldots, Q_k)$ (same number of arguments) and $P_i[\vec{x}] = Q_i[\vec{y}]$ for all $i \in \{1, \ldots, k\}$.

As $P[\overrightarrow{\varphi_A(x)}] \neq Q[\overrightarrow{\varphi_A(y)}]$, there must exist some i such that $P_i[\overrightarrow{\varphi_A(x)}] \neq Q_i[\overrightarrow{\varphi_A(y)}]$. Hence, we can consider smaller contexts P_i and Q_i instead of P and Q.

Consider now the case where the topmost operation of at least one of P and Q is not hashing, i.e. where at least one of the contexts is a variable. W.l.o.g. let Q be a variable. Then there exist $\vec{x}, y \in \mathcal{M}_A \cup \mathcal{M}_{\mathrm{pl}} \cup \mathcal{M}_{A \leftarrow B}$, such that $P[\vec{x}] = y$ (i.e. all messages in $\vec{x}$ are submessages of y), but $P[\overrightarrow{\varphi_A(x)}] \neq \varphi_A(y)$. As this cannot happen if y is an atomic message (else we must have $P = y$ as well, and then $P[\overrightarrow{\varphi_A(x)}] = \varphi_A(y)$), it must be a message received from Bob and $\varphi_A(y) = tr_B(y)$. For such a message y and message context P, we can apply lemma 6 and obtain $P[\overrightarrow{\varphi_A(x)}] = tr_B(y) = \varphi_A(y)$.

Having extended φ_A to a homomorphism, we must show that φ_A is an isomorphism from $\langle \mathcal{M}_A \cdot \mathcal{M}_{\mathrm{pl}} \cdot \mathcal{M}_{A \leftarrow B} \rangle$ to $\langle \mathcal{M}'_A \cdot \mathcal{M}'_{\mathrm{pl}} \cdot \mathcal{M}'_{A \leftarrow B} \rangle$. For this, we have to show that

(i) φ_A maps into $\mathcal{I} = \langle \mathcal{M}'_A \cdot \mathcal{M}'_{\mathrm{pl}} \cdot \mathcal{M}'_{A \leftarrow B} \rangle$;

(ii) φ_A is one-to-one;

(iii) φ_A is onto.

Consider the value of φ_A on $\mathcal{M}_A$, $\mathcal{M}_{\mathrm{pl}}$ and $\mathcal{M}_{A \leftarrow B}$. For nonces x generated by Alice, and for payloads, $\varphi_A(x) = x$, i.e. these values are mapped into $\mathcal{I}$. Consider now a message m received from Bob. In the state $\mathcal{S}^\circ$, or in some earlier state, Bob constructed this message as $P[\vec{y}, \vec{z}, \vec{x}]$ for some

- nonces and messages $\vec{y} \in \mathbf{Y}_B$,
- messages $\vec{z} \in \mathbf{Z}^0$ that Bob received from Alice;
- messages $\vec{x}$ outside $\mathbf{Z}^0$ that Bob received from Alice.

In the state $\mathcal{S}^{\circ\prime}$, Bob constructed the same message as $P[\vec{y}, \vec{z}, \overrightarrow{tr_A(x)}]$. In state $\mathcal{S}^{\bullet\prime}$, this message is

$$tr_B(P[\vec{y}, \vec{z}, \overrightarrow{tr_A(x)}]) = P[\overrightarrow{tr_B(y)}, \vec{z}, \overrightarrow{tr_B(x)}] = \varphi_A(m) \ .$$

Hence φ_A maps m to the corresponding element in $\mathcal{M}'_A \cdot \mathcal{M}'_{\mathrm{pl}} \cdot \mathcal{M}'_{A \leftarrow B}$, meaning that (i) is satisfied. Also, (iii) is satisfied because each of the generators of $\mathcal{I}$ has an original with respect to φ_A.

We still have to show that φ_A is one-to-one. Let $m \in \mathcal{I}$, we show that it has a single preimage by φ_A. If $m = \square^{m'}$ then $\varphi_A^{-1}(m) = m' \in \mathbf{Y}_B$ and there are no other preimages. If m is a nonce in $\mathbf{R}_A$, then it is its own preimage (nothing else is mapped to $\mathbf{R}_A$). If m is a payload, then its preimage is also a payload (uniquely determined). If $m \in \mathbf{Z}^0 \cup \mathbf{C}$, then it is its own preimage by the first case of defining tr_B. In this case, it might be possible that there exists some other m' such that $\varphi_A(m') = m$. Then $m' \notin \mathbf{Z}^0 \cup \mathbf{C}$. The message m' is found as $P[\vec{x}]$ for some message context P and for some messages $\vec{x}$ that are either received from Bob or nonces generated by Alice; $\varphi_A(m')$ is defined by structural induction over P. But Alice's nonces and messages in $\mathbf{Z}^0 \cup \mathbf{C}$ are mapped to themselves. Hence, if $\vec{x}$ contains only such messages, then $\varphi_A(m') = m'$ and $m' = m$. Other possible messages in $\vec{x}$ can be messages received from Bob during the main phase. If they cannot

be generated from Alice's nonces and messages exchanged during the initialization phase, then they must contain some messages in $\mathbf{Y}_B$. But then $\varphi_A(m')$ will contain some $\square^{m''}$ as a submessage and hence $\varphi_A(m') \neq \varphi_A(m)$. □

Lemma 7 shows that as long as Alice and Bob do not send elements of $\mathbf{Z}$ to each other, Eve is able to simulate them to each other. Indeed, Eve knows from the description of the protocol which messages Alice and Bob are going to send to each other — which message context they are applying to their message stores. Although Eve does not necessarily know the message m Alice is sending to Bob, she is capable of constructing the message $tr_A(m)$. But what does Eve do when (say) Alice sends to Bob a tuple of messages $(m_1, \ldots, m_k)$ where $m_i \in \mathbf{Z}$?

In this case, Eve redefines the sets $\mathbf{Z}$ and $\mathbf{Z}^0$, by moving from $\mathbf{Z}$ to $\mathbf{Z}^0$ the message m_i and any other elements of $\mathbf{Z}$ she is now capable of computing. In this way, the mappings tr_A and tr_B are also redefined. Eve now continues as before: applies tr_A to all messages sent by Alice and tr_B to all messages sent by Bob and forwards them to Bob and Alice, respectively. Of course, Bob now notices that Eve is performing an active attack. If Bob had also sent messages belonging to $\mathbf{Z}$, then Alice would have noticed it too.

If the following steps of Alice and Bob do not involve sending messages in the now smaller set $\mathbf{Z}$ to each other, then lemma 7 is again applicable — Eve can simulate Alice to Bob and Bob to Alice. In particular, the following result holds.

Lemma 8. *Let $\cdots \to \mathcal{S}_i \to \mathcal{S}_{i+1} \to \cdots \to \mathcal{S}_j \to \mathcal{S}_{j+1} \to \cdots$ be a trace of the MRP. Let Alice receive the payload M in the state $\mathcal{S}_i$ and let Bob accept M during the step $\mathcal{S}_j \to \mathcal{S}_{j+1}$. If Alice and Bob do not send any elements of $\mathbf{Z}$ to each other during the steps $\mathcal{S}_i \to \mathcal{S}_{i+1}, \ldots, \mathcal{S}_{j-1} \to \mathcal{S}_j$, then the protocol does not have the authenticity property.*

Proof. Eve can simulate the trace $\mathcal{S}_i \to \cdots \to \mathcal{S}_{j+1}$. In particular, she can give a different payload M' to Alice and define $tr_A(M') = M$, $tr_B(M) = M'$. As a result, Bob will still accept M during the step $\mathcal{S}_j \to \mathcal{S}_{j+1}$, but it was never given to Alice to transmit. □

Suppose that the MRP has the perenniality property. Eve's attack against the authenticity of the protocol now works as follows. At the start of the main phase, Eve gives the first payload M_1 to Alice and defines $tr_A(M_1) = M_1'$. As the protocol is perennial, Alice and Bob must exchange messages with the aim of Bob eventually accepting M_1 if Eve has become inactive. During the conversation, Eve uses tr_A and tr_B to rewrite Alice's and Bob's messages, until Bob accepts M_1' (breaking authenticity), or one of the parties includes an element of $\mathbf{Z}$ in the message. So far, Alice and Bob have not noticed the presence of Eve — they have not noticed that they have ended up in the state $\mathcal{S}_i'$ instead of $\mathcal{S}_i$.

Eve now redefines $\mathbf{Z}$ and $\mathbf{Z}^0$, and resets tr_A on payloads (in the following, $tr_A(M) = M$ for an already existing payload M). Alice and Bob now learn that they are in the state $\mathcal{S}_i'$, instead of $\mathcal{S}_i$.

After the redefinition of $\mathbf{Z}$ and $\mathbf{Z}^0$, the premises of lemma 7 are again satisfied. If Alice and Bob continue from the state $\mathcal{S}_i'$ then, as long as they do not send elements of now smaller $\mathbf{Z}$ to each other, they do not notice that Eve applies tr_A to messages sent by Alice and tr_B to messages sent by Bob. In order to make Alice and Bob talk, Eve gives a new payload M_2 to Alice and makes it a non-fixed point of tr_A.

We see that Eve has means to keep Alice and Bob talking. If Bob is accepting messages (which he must do if the protocol has the perenniality property), then Alice and Bob must use up the elements of $\mathbf{Z}$ in order to have authenticity. But the set $\mathbf{Z}$ is finite. Hence, it becomes empty at some point. At this point, Eve knows everything that Alice knows and can masquerade her to Bob.

4.3 Proof for the Full Language

The structure of the proof remains the same. Again, Eve translates the messages exchanged by Alice and Bob such that they do not notice Eve's presence as long as no message from the set $\mathbf{Z}$ is sent (which causes the set $\mathbf{Z}$ to become smaller). Accepting the message by Bob requires consuming an element of $\mathbf{Z}$ which eventually depletes the set and allows Eve to masquerade as Alice. But the definition of $\mathbf{Z}$ is more involved, and we have to be more careful in defining the translation functions tr_A and tr_B.

For a set of messages $\mathbf{M}$, define its *linear hull* as $\langle\langle \mathbf{M} \rangle\rangle = \{m \in \Sigma \mid \mathbf{M} \vdash_{\oplus} m\}$. Let $\mathbf{W}^0$ be the set of all messages that Alice and Bob have sent during the initialization phase. Let $\mathbf{W}$ be the set containing $\mathbf{W}^0$, as well as all submessages of messages in $\mathbf{W}^0$. Let $\mathbf{Z}^0$ contain all messages in $\langle\langle \mathbf{W} \rangle\rangle$ that are known to Eve at the end of the initialization phase. Let $\mathbf{Z} = \langle\langle \mathbf{W} \rangle\rangle \backslash \mathbf{Z}^0$. As before, $\mathbf{Z}_A \subseteq \mathbf{Z}$ is the set of messages in $\mathbf{Z}$ known only to Alice, and $\mathbf{Z}_B = \mathbf{Z} \backslash \mathbf{Z}_A$ is the set of messages in $\mathbf{Z}$ known only to Bob. As before, let $\mathbf{Y}_A = \mathbf{Z}_A \cup \mathbf{R}_A$ and $\mathbf{Y}_B = \mathbf{Z}_B \cup \mathbf{R}_B$. We now define $tr_A(m)$ and $tr_B(m)$ as follows:

$$tr_A(m) = \begin{cases} m, & m \in \mathbf{Z}^0 \cup \mathbf{C} \\ \square^m, & m \in \mathbf{Y}_A \\ m', & m = \square^{m'}, m' \in \mathbf{Y}_B \\ tr_A(m_1) \oplus \cdots \oplus tr_A(m_k), & m = m_1 \oplus \cdots \oplus m_k \\ h(tr_A(m_1), \ldots, tr_A(m_k)), & m = h(m_1, \ldots, m_k) \end{cases}$$

$$tr_B(m) = \begin{cases} m, & m \in \mathbf{Z}^0 \cup \mathbf{C} \\ \square^m, & m \in \mathbf{Y}_B \\ m', & m = \square^{m'}, m' \in \mathbf{Y}_A \\ tr_B(m_1) \oplus \cdots \oplus tr_B(m_k), & m = m_1 \oplus \cdots \oplus m_k \\ h(tr_B(m_1), \ldots, tr_B(m_k)), & m = h(m_1, \ldots, m_k) \end{cases}$$

Again, the cases must be considered from top to bottom.

Eve performs the attack by replacing messages m sent by Alice with $tr_A(m)$ and messages m sent by Bob with $tr_B(m)$, *as long as it is possible*. In Sec. 4.2, the condition "as long as it is possible" was very simple — the messages sent by Alice and Bob could not contain elements of $\mathbf{Z}$. With XOR, the condition is somewhat more complicated — the messages in $\mathbf{Z}^0$, together with the messages sent by Alice and Bob during the main phase should not allow Eve to find any message in $\mathbf{Z}$.

We have an analogue of Lemma 5 stating that Eve can actually perform the replacement of messages m with $tr_A(m)$ or $tr_B(m)$. The proof carries over without any changes.

We have to prove an analogue for Lemma 7. We have the same situation as before — there are two global states $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$. In both of these states, $\mathbf{W}^0$ is a set of messages

that are known to both Alice and Bob (and hence also Eve), $\mathbf{W}$ contains all messages in $\mathbf{W}^0$ and their submessages, $\mathbf{Z}^0$ is the set of messages in $\langle\langle \mathbf{W} \rangle\rangle$ that are known to both Alice and Bob (and Eve), and $\mathbf{Z} = \langle\langle \mathbf{W} \rangle\rangle \backslash \mathbf{Z}^0$. In states $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$, Alice's and Bob's internal states are equal. Also, Alice's views in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ are related by an isomorphism φ_A that is the identity on nonces in $\mathbf{R}_A$ and equals tr_B on messages received from Bob and on payloads. Similarly, Bob's views in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ are related by an isomorphism φ_B that is the identity on nonces in $\mathbf{R}_B$ and equals tr_A on messages received from Alice. We consider steps $\mathcal{S}^\circ \to \mathcal{S}^\bullet$ and $\mathcal{S}^{\circ\prime} \to \mathcal{S}^{\bullet\prime}$. In the first case, Eve is passive, and in the second, Eve applies tr_A to messages sent by Alice and tr_B to messages sent by Bob. But first we extend Lemma 6.

LEMMA 6'. *Let $\mathbf{Z}$, $\mathbf{Z}^0$ and tr_B be defined as above. Let $\mathcal{M}$ be a message store of Alice. Let φ be a mapping on messages defined as in Lemma 6. Let y be a message satisfying the following:*

- *From y, messages in $\mathbf{Z}^0$, and messages received from Bob (in $\mathcal{M}$), it is impossible to derive any message in $\mathbf{Z}$.*
- *Adding y to the knowledge of Alice does not allow her to compute any more messages in $\mathbf{Z}$ compared to what she can compute just from $\mathcal{M}$.*
- *Second condition in Lemma 6: If $r \in \mathbf{Y}_A$ and $r \sqsubseteq y$, then exists $z \in \mathbf{Z}^0$, such that $r \sqsubseteq z \sqsubseteq y$. Such z exists for each occurrence of r in y.*

Let $P[\vec{x}] = y$ for a certain message context P and messages $\vec{x}$ in $\mathcal{M}$. Then $P[\overrightarrow{\varphi(x)}] = tr_B(y)$.

Proof. We note that here the message y may be just another message that Alice has received (and then $\varphi(x) = tr_B(x)$). Hence, we assume that y is really an element of $\mathcal{M}$ and prove by induction over the sum of the sizes of contexts P and Q the statement "If $P[\vec{x}] = Q[\vec{x}]$, then $P[\overrightarrow{\varphi(x)}] = Q[\overrightarrow{\varphi(x)}]$".

Base: P and Q are variables. Then they must point to the same message.

Step: We consider the possible shapes of P and Q. If $P = h(P_1, \ldots, P_k)$ and Q is a variable y (an element of $\vec{x}$), then the message y must also be a formal hash and hence received from Bob. Similarly to the proof of Lemma 6, we consider which case was used to define $tr_B(y)$. If $y \in \mathbf{Z}^0 \cup \mathbf{C}$, then $tr_B(y) = y$ and for all elements x of the vector $\vec{x}$ that actually occur in the context, $\varphi(x) = x$. If $tr_B(y)$ was defined inductively, then the induction assumption gives us $P_i[\overrightarrow{\varphi(x)}] = tr_B(y_i)$ for all $i \in \{1, \ldots, k\}$. Here $y = h(y_1, \ldots, y_k)$. Note that for the induction step, we can add $y_1, \ldots, y_k$ to $\mathcal{M}$. They do not allow Alice the computation of any more elements of $\mathbf{Z}$, because she can already compute y_i as $P_i[\vec{x}]$.

If $P = P_1 \oplus \cdots \oplus P_k$ (where each P_i is a variable of has hashing as the outermost operation) and Q is the constant 0, then let us define H as the set of messages where

- each element of H is a formal hash or a nonce;
- for all message contexts P_i, if the topmost constructor of P_i is hashing, then $P_i[\vec{x}] \in H$;

- for all contexts P_i, if P_i is a variable x_j and the message corresponding to this variable is $m_1 \oplus \cdots \oplus m_r$, where $m_1, \ldots, m_r$ are formal hashes or nonces, then $m_1, \ldots, m_r \in H$.

That is, the elements of H are the immediate submessages of $P[\vec{x}]$ after flattening the outermost $\oplus$-operations. Each $m \in H$ is caused to be in H by an even number of contexts P_i (we assume that the context P has been simplified as much as possible).

If there is some P_i whose topmost operation is hashing, then let $m_i = P_i[\vec{x}]$. By the induction assumption, $P_i[\overrightarrow{\varphi(x)}] = \varphi(m_i) = tr_B(m_i)$ (we may add m_i to $\mathcal{M}$). Also, let $P' = P \oplus P_i \oplus v$, where v is a new variable. The size of P' is smaller than the size of P (the components P_i cancel out). We have $P'[\vec{x}, m_i] = 0$ (the message m_i is assigned to the new variable v). Again by induction assumption, $P'[\overrightarrow{\varphi(x)}, \varphi(m_i)] = \varphi(0) = 0$. Combining it with $P_i[\overrightarrow{\varphi(x)}] = \varphi(m_i)$ gives us $P[\overrightarrow{\varphi(x)}] = 0$.

If $P_1, \ldots, P_k$ are all variables, then let $m_1, \ldots, m_k$ be the messages in $\mathcal{M}$ assigned to them. We have $m_i = \bigoplus H_i$, where $H_i \subseteq H$ is the set of nonces and formal hashes whose XOR is m_i. We know that $m_1 \oplus \cdots \oplus m_k = 0$ and must show $\varphi(m_1) \oplus \cdots \oplus \varphi(m_k) = 0$. If all messages m_i are received from Bob, then $\varphi(m_i) = tr_B(m_i) = \bigoplus_{z \in H_i} tr_B(z)$ and they cancel out when XOR'ed. If some m_i is a nonce $r \in \mathbf{R}_A$ generated by Alice, then the same r must also occur in some other message m_j from Bob. But in this case, it is possible to find either r or some XOR of Alice's nonces from the messages in $\mathbf{Z}^0$ and the messages received from Bob. This is impossible according to the premises of the lemma. We have handled the case $P = P_1 \oplus \cdots \oplus P_k$ and $Q = 0$.

If $P = h(P_1, \ldots, P_k)$ and $Q = h(Q_1, \ldots, Q_k)$, then we apply induction assumption to each P_i and Q_i. If $P = P_1 \oplus \cdots \oplus P_k$ and $Q \neq 0$, then we apply the induction assumption to the contexts $P' = P \oplus Q$ and 0. □

LEMMA 7'. *If the messages sent by Alice and Bob in $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$ do not allow Eve to deduce elements of $\mathbf{Z}$, then in states $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$, the messages in the set $\mathbf{Z}$ are still known by exactly one of Alice and Bob. Furthermore, $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$ are isomorphic, and the isomorphism between messages is related to tr_A and tr_B in the same way as for $\mathcal{S}^\circ$ and $\mathcal{S}^{\circ\prime}$.*

Here "messages sent by Alice and Bob" that Eve may consider to deduce elements of $\mathbf{Z}$ include all messages in Alice's store that are not nonces created by Alice, and all messages in Bob's store that are not nonces created by Bob.

Proof. As before, Alice's internal state in $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$ are the same. Also, Bob's internal state is the same, and Alice and Bob do not learn any messages in $\mathbf{Z}$.

Next, consider again the extension of φ_A to a homomorphism of Alice's views in $\mathcal{S}^\bullet$ and $\mathcal{S}^{\bullet\prime}$. The mapping φ_A is defined on the components $\mathcal{M}_A \cdot \mathcal{M}_{\mathrm{pl}} \cdot \mathcal{M}_{A \leftarrow B}$ of $\mathcal{S}^\bullet$ by $\varphi_A(x) = x$ for $x \in \mathbf{R}_A$ and $\varphi_A(m) = tr_B(m)$ for other messages. The possibility to extend φ_A to homomorphism follows directly from Lemma 6'.

We also have to show that φ_A is an isomorphism from $\langle \mathcal{M}_A \cdot \mathcal{M}_{\mathrm{pl}} \cdot \mathcal{M}_{A \leftarrow B} \rangle$ to $\mathcal{I} = \langle \mathcal{M}'_A \cdot \mathcal{M}'_{\mathrm{pl}} \cdot \mathcal{M}'_{A \leftarrow B} \rangle$. The proof that φ_A maps into $\mathcal{I}$ and onto $\mathcal{I}$ remains the same. Also, showing that φ_A is one-to-one proceeds exactly as before by showing that each $m \in \mathcal{I}$ has just a single preimage. □

Having proved the simulatability Lemma 7', the rest of the proof proceeds exactly as in Sec. 4.2. Eve translates the messages as long as no elements of $\mathbf{Z}$ (or messages that allow the computation of elements of $\mathbf{Z}$) are exchanged between Alice and Bob. Should that happen, Eve redefines the sets $\mathbf{Z}^0$ and $\mathbf{Z}$ and continues with the simulation. Alice and Bob must keep talking (because of the perenniality property) and opening elements of $\mathbf{Z}$. Eventually, the finite set $\mathbf{Z}$ will be depleted and Eve can masquerade as Alice.

5 Conclusions

With advances in hardware, asymmetric cryptography is fast becoming a viable option for small devices. For example, in wireless sensor networks, Claycomb et al. propose a key-establishment protocol using group-based techniques combined with identity-based cryptography [8]. In [16], Liu and Ning propose using TinyECC, a configurable library in wireless sensor networks using elliptic curve cryptography. Nevertheless, we believe that there will always exist devices, ever smaller, with computational capabilities similar to the least powerful devices of today. Our results show that for these devices, certain forms of authentication are impossible.

We have shown that for a certain set of cryptographic primitives, the perennial authentication is impossible. An interesting future work, complementing [23] would be the determination of necessary and/or sufficient properties on symbolic cryptographic primitives for the possibility of authentication.

Our result has been established in the symbolic setting. Interestingly, it does *not* hold in the computational setting where signature schemes can be constructed from symmetric encryption [20] and one-way hash functions. This points out a gap between the two models, which according to our knowledge has not been recognized before. It would be interesting to study the gap and find out methods to reduce it, thereby finding symbolic model that better capture the essentials of cryptography. Regarding Merkle's construction, one of its main tools is decomposing messages to their constituent bits. The introduction of bits to the symbolic model is well-known to be very difficult [15].

References

[1] M. Abadi and M. R. Tuttle. A Semantics for a Logic of Authentication (Extended Abstract). In *PODC*, pages 201–216, 1991.

[2] R. Anderson, F. Bergadano, B. Crispo, J.-H. Lee, C. Manifavas, and R. Needham. A New Family of Authentication Protocols. *Operating Systems Review*, 32(4):9–20, 1998.

[3] M. Backes and B. Pfitzmann. Limits of the Cryptographic Realization of Dolev-Yao-Style XOR. In S. De Capitani di Vimercati, P. Syverson, and D. Gollmann, editors, *Computer Security ESORICS 2005: 10th European Symposium on Research in Computer Security*, volume 3679 of *Lecture Notes in Computer Science*, pages 178–196, Berlin, Heidelberg, 2005. Springer.

[4] M. Backes, B. Pfitzmann, and M. Waidner. Limits of the BRSIM/UC Soundness of Dolev-Yao Models with Hashes. In E. Asarin, D. Gollmann, J. Meier, and A. Sabelfeld, editors, *Computer Security ESORICS 2006: 11th European Symposium on Research in Computer Security*, volume 4189 of *Lecture Notes in Computer Science*, pages 404–423, Berlin, Heidelberg, 2006. Springer.

[5] A. Buldas and A. Jürgenson. Does Secure Time-Stamping Imply Collision-Free Hash Functions? In W. Susilo, J. K. Liu, and Y. Mu, editors, *ProvSec*, volume 4784 of *Lecture Notes in Computer Science*, pages 138–150. Springer, 2007.

[6] A. Buldas and M. Niitsoo. Can We Construct Unbounded Time-Stamping Schemes from Collision-Free Hash Functions? In J. Baek, F. Bao, K. Chen, and X. Lai, editors, *ProvSec*, volume 5324 of *Lecture Notes in Computer Science*, pages 254–267. Springer, 2008.

[7] A. Buldas and M. Saarepera. On Provably Secure Time-Stamping Schemes. In P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 500–514. Springer, 2004.

[8] W. R. Claycomb, R. Lopes, D. Shin, and B. Kim. Key Establishment Using Group Information for Wireless Sensor Networks. In S. Hailes, S. Sicari, and G. Roussos, editors, *Sensor Systems and Software*, volume 24 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 51–65. Springer Berlin Heidelberg, 2010.

[9] D. Dolev and A. C.-C. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[10] O. Goldreich. *Foundations of Cryptography, Volume I - Basic Techniques*. Cambridge University Press, 2001.

[11] M. González Muñiz and R. Steinwandt. Cryptanalysis of a Message Recognition Protocol by Mashatan and Stinson. In *ICISC '09: 12th International Conference on Information Security and Cryptology*, 2009.

[12] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *STOC '89: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 44–61, New York, NY, USA, 1989. ACM.

[13] P. Laud. Implementing Cryptographic Primitives in the Symbolic Model. In M. Bobaru, K. Havelund, G. Holzmann, and R. Joshi, editors, *Third NASA Formal Methods Symposium*, volume 6617 of *Lecture Notes in Computer Science*, pages 267–281. Springer Berlin / Heidelberg, 2011.

[14] S. Laur and S. Pasini. User-Aided Data Authentication. *International Journal of Security and Networks*, 4(1/2):69–86, 2009.

[15] P. Lincoln, J. C. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 112–121, 1998.

[16] A. Liu and P. Ning. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *IPSN '08: Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, pages 245–256, Washington, DC, USA, 2008. IEEE Computer Society.

[17] S. Lucks, E. Zenner, A. Weimerskirch, and D. Westhoff. Concrete Security for Entity Recognition: The Jane Doe Protocol. In D. R. Chowdhury, V. Rijmen, and A. Das, editors, *Progress in Cryptology – INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 158–171. Springer-Verlag, 2008.

[18] A. Mashatan and D. R. Stinson. A New Message Recognition Protocol for Ad Hoc Pervasive Networks. In M. K. Franklin, L. C. K. Hui, and D. S. Wong, editors, *Cryptology and Network Security, 7th International Conference, CANS 2008*, volume 5339 of *Lecture Notes in Computer Science*, pages 378–394. Springer-Verlag, 2008.

[19] A. Mashatan, D. R. Stinson, and I. Goldberg. A New Message Recognition Protocol with Self-recoverability for Ad Hoc Pervasive Networks. In M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, editors, *Applied Cryptography and Network Security*, volume 5536 of *Lecture Notes in Computer Science*, pages 219–237. Springer Berlin / Heidelberg, 2009.

[20] R. C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In C. Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 1987.

[21] C. J. Mitchell. Remote User Authentication Using Public Information. In K. G. Paterson, editor, *Cryptography and Coding, 9th IMA International Conference*, volume 2398 of *Lecture Notes in Computer Science*, pages 360–369. Springer-Verlag, 2003.

[22] O. Pereira and J.-J. Quisquater. On the Impossibility of Building Secure Cliques-Type Authenticated Group Key Agreement Protocols. *Journal of Computer Security*, 14(2):197–246, 2006.

[23] B. Schmidt, P. Schaller, and D. Basin. Impossibility Results for Secret Establishment. In *CSF*. IEEE Computer Society, 2010. To appear.

[24] D. R. Simon. Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In *EUROCRYPT*, pages 334–345, 1998.

[25] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, editors, *Security Protocols, 7th International Workshop*, volume 1796 of *Lecture Notes in Computer Science*, pages 172–182. Springer-Verlag, 2000.

[26] A. Weimerskirch and D. Westhoff. Zero Common-Knowledge Authentication for Pervasive Networks. In M. Matsui and R. Zuccherato, editors, *Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 73–87. Springer-Verlag, 2004.

[27] J. Zhou and D. Gollmann. A Fair Non-repudiation Protocol. In *IEEE Symposium on Security and Privacy*, pages 55–61. IEEE Computer Society, 1996.