Completeness of Symbolic Hashes in the Standard Model

Flavio D. Garcia and Peter van Rossum

Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands.
{flaviog,petervr}@cs.ru.nl

Abstract. We study an extension of the well-known Abadi-Rogaway logic with hashes. Previously, we have shown that it is possible to give a sound computational interpretation of this extension in the standard model using Canetti's oracle hashing. This paper extends Micciancio and Warinschi's completeness result for the original logic to this setting.

1 Introduction

The analysis of security protocols is being carried out mainly by means of two different techniques. On the one hand, there is the logic approach, which sees messages as algebraic objects defined using some formal language. In this view, cryptographic operations are algebraic operations which are unbreakable. Attackers are typically modelled as so-called Dolev-Yao attackers [DY83], having total control over the network, having no computational limitations, and being only (but absolutely) incapable of breaking cryptographic operations. This view is appealing, because it is relatively easy to use and captures most mistakes commonly made in security protocols.

On the other hand, there is the complexity-based approach. Here messages are bit strings and cryptographic operations are functions on bit strings satisfying certain security properties [Gol01]. Common security notions like secrecy, authenticity, and integrity are formulated 'in terms of the probability that someone can mount a successful attack. An attacker here is a resource bounded probabilistic algorithm, limited by running time and/or memory, but capable of breaking cryptographic operations, if that is computationally feasible. The complexity based methods are more general and more realistic, but also more complex to use.

In the last few years much research has been done to relate these two perspectives [AR02,AJ01,MW04,Her05]. Such a relation takes the form of a function mapping algebraic messages m to (distributions over) bit strings $[\![m]\!]$. This map then should relate messages that are observationally equivalent in the algebraic world (meaning that a Dolev-Yao attacker can see no difference between them) to indistinguishable distributions over bit strings (meaning that a computationally bounded adversary can only with negligible probability distinguish the distributions). Such a map allows one to use algebraic methods, possibly even automated, to reason about security properties of protocols and have those reasonings be valid also in the computational world.

Recently, some study has been made on extending the Abadi-Rogaway logic and the mapping to the computational world with hashes [GvR06b,GvR06c,GvR06a]. In fact, in

the conclusions of [MW04], Micciancio and Warinschi briefly but explicitly question if this logical approach can be extended to, among other things, collision resistant hashes. Backes, Pfitzmann, and Waidner [BPW06] show that in their simulatability framework [PW00] a sound interpretation of hashes cannot exist, but that it is possible to give a sound interpretation of formal hashes in the simulatability framework using random oracles.

The problem with hashes is that in the algebraic world $h(m)$ and $h(m')$ are indistinguishable for a Dolev-Yao attacker if the attacker does not know m and m'. In the computational world, however, the standard security definition — it must be computationally infeasible to compute any pre-image of a hash value or a hash collision [RS04] — does not guarantee that the hash function hides all partial information about the message; hence there is no guarantee that $[\![h(m)]\!]$ and $[\![h(m')]\!]$ are computationally indistinguishable.

In [GvR06c], the present authors give a sound interpretation of formal hashes using the notion of perfectly one-way functions (a.k.a. oracle hashing) from Canetti and others [Can97,CMR98]. These functions are probabilistic hashes that hide all partial information.

Completeness. Although soundness results allow us to port proofs of *secrecy* properties from the algebraic world to the computation world, it does not permit to port, for instance, *authenticity* and *integrity* results. For hashes, this limits the usefulness of a soundness result. For example, consider a protocol in which an agent A chooses a nonce n and commits to this nonce by sending $h(n)$ to another agent B. Later in the protocol, A will reveal the nonce n by sending n itself to B. Security in this setting means that A cannot change her choice after sending $h(n)$. In the algebraic world, this is guaranteed by the fact that the message $h(n)n$ (the concatenation of the relevant messages in the protocol run) is observationally distinct from $h(n)n'$, with $n' \neq n$. We would like to be able to conclude from this algebraic property that $[\![h(n)n]\!]$ is computationally distinct from $[\![h(n)n']\!]$, since that is needed to guarantee the security in the computational world.

What is needed here is completeness: computational equivalence of $[\![m]\!]$ and $[\![m']\!]$ should imply observational equivalence of m and m'. For the original Abadi-Rogaway logic, completeness under appropriate conditions on the encryption scheme was proven by Micciancio and Warinschi [MW04]. In this paper we extend this completeness proof to our version with hashes.

Overview. Section 2 introduces the message algebra from [GvR06c], including the probabilistic encryption and probabilistic hash operators. It also defines the observational equivalence relation on messages. Section 3 then introduces the computational world, giving the security definitions for encryption and hashes. In Section 4 the semantic interpretation $[\![-]\!]$ is defined and the soundness result of [GvR06c] is recalled. Section 5 proves the completeness of the interpretation. The proof presented here proceeds along the lines of [MW04], reusing results from that paper for simplicity whenever feasible.

2 The algebraic setting

This section describes the message space and the observational equivalence from [GvR06c] extending the well known Abadi-Rogaway logic [AR02] of algebraic messages with hashes.

These messages are used to describe cryptographic protocols and the observational equivalence tells whether or not two protocol runs are indistinguishable for a global eavesdropper. This setting is exactly the same as in [GvR06c], so here we only give a brief summary.

Definition 2.1. *Messages* are constructed using algebraic encryption, hashing, and pairing operations from keys ($k \in \mathrm{Key}$), nonces ($n \in \mathrm{Nonce}$), randomness labels ($r \in \mathrm{Random}$), and constants ($c \in \mathrm{Const}$):

$$\mathrm{Msg} \ni m := c \mid k \mid n \mid \{\!|m|\!\}_k^r \mid \mathrm{h}^r(m) \mid \langle m, m\rangle \mid \square^r \mid \boxtimes^r .$$

There is one special key called $k_\square$ and for every randomness label r there is a special nonce called $n_\boxtimes^r$; these are only used when interpreting the special symbols ($\square^r$ and $\boxtimes^r$) as bit string and do not otherwise form a part of the message algebra.

The *closure* of a set U of messages is the set of all messages that can be constructed from U using tupling, detupling, and decryption. It represents the information an adversary could deduce knowing U.

Definition 2.2 (Closure). The *closure* of a set U of messages, denoted by $\overline{U}$, is the smallest set of messages satisfying: 1. $\mathrm{Const} \subseteq \overline{U}$; 2. $U \subseteq \overline{U}$; 3. $m, m' \in \overline{U} \implies \langle m, m'\rangle \in \overline{U}$; 4. $\{\!|m|\!\}_k^r, k \in \overline{U} \implies m \in \overline{U}$; 5. $\langle m, m'\rangle \in \overline{U} \implies m, m' \in \overline{U}$. For the singleton set $\{m\}$, we write $\overline{m}$ instead of $\overline{\{m\}}$.

The function *pattern*: $\mathrm{Msg} \to \mathrm{Msg}$ from [GvR06c] is a straightforward extension from the same function in Abadi-Rogaway [AR02] which takes a message m and reduces it to a pattern. Intuitively, this is the pattern that an attacker sees in a message given that she knows the messages in U.

$$\begin{aligned}
\mathit{pattern}(m) &= \mathit{pattern}(m, \overline{m}) \\
\mathit{pattern}(\langle m_1, m_2\rangle, U) &= \langle \mathit{pattern}(m_1, U), \mathit{pattern}(m_2, U)\rangle \\
\mathit{pattern}(\{\!|m|\!\}_k^r, U) &= \begin{cases} \{\!|\mathit{pattern}(m, U)|\!\}_k^r, & \text{if } k \in U; \\ \square^{\mathcal{R}(\{\!|m|\!\}_k^r)}, & \text{otherwise.} \end{cases} \\
\mathit{pattern}(\mathrm{h}^r(m), U) &= \begin{cases} \mathrm{h}^r(\mathit{pattern}(m, U)), & \text{if } m \in U; \\ \boxtimes^{\mathcal{R}(\mathrm{h}^r(m))}, & \text{otherwise.} \end{cases} \\
\mathit{pattern}(m, U) &= m \quad \text{in any other case.}
\end{aligned}$$

Here $\mathcal{R}\colon \mathrm{Enc} \cup \mathrm{Hash} \hookrightarrow \mathrm{Random}$ is an injective function that takes an encryption or a hash value and outputs a tag that identifies its randomness. This tagging function is needed to make sure that the function *pattern* is injective: distinct undecryptable messages should be replaced by distinct boxes and similarly for hashes.

Example 2.3. Consider the message

$$m = \langle \{\!|\{\!|1|\!\}_{k'}^{r'}, \mathrm{h}^{\tilde{r}}(n)|\!\}_k^r, \mathrm{h}^{\hat{r}}(k), k\rangle.$$

Then $\mathit{pattern}(m) = \langle \{\!|\ \square^s\ ,\ \boxtimes^t\ |\!\}_k^r, \mathrm{h}^{\hat{r}}(k), k\rangle$, because k', n are not in $\overline{m}$,

where $t = \mathcal{R}(\mathrm{h}^{\tilde{r}}(n))$ and $s = \mathcal{R}(\{\!|1|\!\}_{k'}^{r'})$.

Definition 2.4 (Renaming). Two messages m and m' are said to be *equivalent up to renaming*, notation $m \equiv m'$, if there is a type preserving permutation σ of $\mathrm{Key} \cup \mathrm{Nonce} \cup \mathrm{Box} \cup \mathrm{Random}$ such that $m = m'\sigma$. Here $m'\sigma$ denotes simultaneous substitution of x by $\sigma(x)$ in m', for all $x \in \mathrm{Key} \cup \mathrm{Nonce} \cup \mathrm{Box} \cup \mathrm{Random}$.

Definition 2.5 (Observational equivalence). Two messages m and m' are said to be *observationally equivalent*, denoted by $m \cong m'$, if $\mathit{pattern}(m) \equiv \mathit{pattern}(m')$.

From the original setting in [AR02] we inherit the requirement that messages must be acyclic.

Definition 2.6 (Acyclicity). Let m be a message and k, k' two keys. The key k is said to *encrypt* k' *in* m if m has a sub-message of the form $\{\!|m'|\!\}_k^r$ with k' being a sub-message of m'. A message is said to be *acyclic* if there is no sequence $k_1, k_2, \ldots, k_n, k_{n+1} = k_1$ of keys such that k_i encrypts k_{i+1} in m for all $i \in \{1, \ldots, n\}$.

3 The computational setting

This section gives a brief overview of the concepts used in the complexity theoretic approach to security protocols. Much of this is standard; the reader is referred to [GB01,BDJR97] for a thorough treatment of the basic concepts, to [AR02] for the notion of *type-0 security* for encryption schemes (see Section 3.1 below), and to [Can97] for the notion of *oracle hashing* (see Section 3.2 below). This section follows the same lines as [GvR06c]; here we also focus on the extra requirements needed to achieve completeness: confusion freeness for encryption schemes (Definition 3.3) and collision resistance for hash schemes (Definition 3.5).

In the computational world, messages are elements of $\mathrm{Str} := \{0,1\}^*$. Cryptographic algorithms and adversaries are probabilistic polynomial-time algorithms. When analyzing cryptographic primitives, it is customary to consider probabilistic algorithms that take an element in $\mathrm{Param} := \{1\}^*$ as input, whose length scales with the security parameter. By making the security parameter large enough, the system should become arbitrarily hard to break.

This idea is formalized in the security notions of the cryptographic operations. The basic one, which is what is used to define the notion of semantically equivalent messages, is that of *computational indistinguishability* of probability ensembles over Str. Here a *probability ensemble over* Str is a sequence $\{A_\eta\}_{\eta \in \mathbb{N}}$ of probability distributions over Str indexed by the security parameter.

Definition 3.1 (Computational indistinguishability). Two probability ensembles $\{X_\eta\}_\eta$ and $\{Y_\eta\}_\eta$ are *computationally indistinguishable* if for every probabilistic polynomial-time algorithm D,

$$\mathbb{P}[x \xleftarrow{\$} X_\eta; D(1^\eta, x) = 1] - \mathbb{P}[x \xleftarrow{\$} Y_\eta; D(1^\eta, x) = 1]$$

is a negligible function of η.

Recall that a function $f\colon \mathbb{N} \to \mathbb{N}$ is called *negligible* if for all positive polynomials p, $f(\eta) \leq \frac{1}{p(\eta)}$ for large enough η. We now give the formal definition of an encryption scheme and its security notion in Section 3.1 and of oracle hashing in Section 3.2.

3.1 Encryption scheme

For each security parameter $\eta \in \mathbb{N}$ we let $\text{Plaintext}_\eta \subseteq \text{Str}$ be a non-empty set of *plaintexts*, satisfying that for each $\eta \in \mathbb{N}$: $\text{Plaintext}_\eta \subseteq \text{Plaintext}_{\eta+1}$ as in Goldwasser and Bellare [GB01]. Let us define $\text{Plaintext} = \bigcup_\eta \text{Plaintext}_\eta$. There is a set $\text{Keys} \subseteq \text{Str}$ of *keys* and also a set $\text{Ciphertext} \subseteq \text{Str}$ of *ciphertexts*. Furthermore, there is a special bit string $\perp$ not appearing in Plaintext or Ciphertext. An *encryption scheme* Π consists of three algorithms:

1. a (probabilistic) key generation algorithm $\mathcal{K}\colon \text{Param} \to \text{Keys}$ that outputs, given a unary sequence of length η, a randomly chosen element of Keys;
2. a (probabilistic) encryption algorithm $\mathcal{E}\colon \text{Keys} \times \text{Str} \to \text{Ciphertext} \cup \{\perp\}$ that outputs, given a key and a bit string, a possibly randomly chosen element from Ciphertext or $\perp$;
3. a (deterministic) decryption algorithm $\mathcal{D}\colon \text{Keys} \times \text{Str} \to \text{Plaintext} \cup \{\perp\}$ that outputs, given a key and a ciphertext, an element from Plaintext or $\perp$.

These algorithms must satisfy that the decryption (with the correct key) of a ciphertext returns the original plaintext.

Now we define type-0 security of an encryption scheme as in [AR02], which is a variant of the standard semantic security definition, enhanced with some extra properties. In particular a type-0 secure encryption scheme is which-key concealing, repetition concealing and length hiding. We refer to the original paper for motivation and explanations on how to achieve such an encryption scheme.

Definition 3.2. An *adversary (for type-0 security)* is a probabilistic polynomial-time algorithm $A^{\mathcal{F}(-),\mathcal{G}(-)}\colon \text{Param} \to \{0,1\}$ having access to two probabilistic oracles $\mathcal{F}, \mathcal{G}\colon \text{Str} \to \text{Str}$. The *advantage* of such an adversary is the function $\text{Adv}_A\colon \mathbb{N} \to \mathbb{R}$ defined by

$$\text{Adv}_A(\eta) = \mathbb{P}[\kappa, \kappa' \xleftarrow{\$} \mathcal{K}(1^\eta); A^{\mathcal{E}(\kappa,-),\mathcal{E}(\kappa',-)}(1^\eta) = 1] - \mathbb{P}[\kappa \xleftarrow{\$} \mathcal{K}(1^\eta); A^{\mathcal{E}(\kappa,0),\mathcal{E}(\kappa,0)}(1^\eta) = 1].$$

Here the probabilities are taken over the choice of κ and κ' by the key generation algorithm, over the choices of the oracles, and over the internal choices of A. An encryption scheme $\langle \mathcal{K}, \mathcal{E}, \mathcal{D} \rangle$ is called *type-0 secure* if for all polynomial-time adversaries A as above, the advantage Adv_A is a negligible function of η.

For completeness it is needed that the decryption algorithm returns *reject* whenever it is called with a key that does was not used to encrypt the message in the first place. The special bit string $\perp$ is used to indicate failure of decryption. This property is called *confusion freeness*. See [MW04], where the completeness for the original Abadi-Rogaway logic is proven.

Definition 3.3 (Confusion freeness). Let $\Pi = \langle \mathcal{K}, \mathcal{E}, \mathcal{D} \rangle$ be an encryption scheme indexed by the security parameter η. Π is said to be *confusion free* if for all bit strings μ the probability

$$\mathbb{P}[\kappa_1, \kappa_2 \xleftarrow{\$} \mathcal{K}(\eta) : \mathcal{D}_{\kappa_1}(\mathcal{E}_{\kappa_2}(\mu)) \neq \perp]$$

is a negligible function of η.

3.2 Oracle hashing

In [GvR06c], Canetti's notion of *oracle hashing* [Can97] is used as the computational counterpart to the algebraic hash operation. A *hash scheme* consists of two algorithms $\mathcal{H}$ and $\mathcal{V}$. The probabilistic algorithm $\mathcal{H}\colon \text{Param} \times \text{Str} \to \text{Str}$ takes a unary sequence and a message and outputs a hash value; the verification algorithm $\mathcal{V}\colon \text{Str} \times \text{Str} \to \{0,1\}$ that given two messages x and c correctly decides whether c is a hash of x or not.

Canetti gives essentially two security notions for such a hash scheme. The first one, *oracle indistinguishability*, guarantees that an adversary can gain no information at all about a bit string, given its hash value (or rather, with sufficiently small probability). This property is used in [GvR06c] to prove the soundness of the interpretation. The second one is an appropriate form of collision resistance. It guarantees that an adversary cannot (or rather, again, with sufficiently small probability) compute two distinct messages that successfully pass the verification test with the same hash value. This property will be used in this paper to prove completeness.

Definition 3.4 (Oracle indistinguishability). A hash scheme $\langle \mathcal{H}, \mathcal{V} \rangle$ is said to be *oracle indistinguishable* if for every family of probabilistic polynomial-time predicates $\{D_\eta\colon \text{Str} \to \{0,1\}\}_{\eta \in \mathbb{N}}$ and every positive polynomial p there is a polynomial size family $\{L_\eta\}_{\eta \in \mathbb{N}}$ of subsets of Str such that for all large enough η and all $x, y \in \text{Str} \setminus L_\eta$:

$$\mathbb{P}[D_\eta(\mathcal{H}(1^\eta, x)) = 1] - \mathbb{P}[D_\eta(\mathcal{H}(1^\eta, y)) = 1] < \frac{1}{p(\eta)}.$$

Here the probabilities are taken over the choices made by $\mathcal{H}$ and the choices made by D_η.

Definition 3.5 (Collision resistance). A hash scheme $\langle \mathcal{H}, \mathcal{V} \rangle$ is said to be *collision resistant* if for every probabilistic polynomial time adversary A, the probability

$$\mathbb{P}[\langle c, x, y \rangle \xleftarrow{\$} A(1^\eta); x \neq y \wedge \mathcal{V}(x, c) = \mathcal{V}(y, c) = 1]$$

is a negligible function of η.

As an example we reproduce here a hash scheme proposed in [Can97] that satisfies both security notions. Let p be a large (i.e., scaling with η) safe prime. Take $\mathcal{H}(x) = \langle r^2, r^{2 \cdot h(x)} \mod p \rangle$, where r is a randomly chosen element in $\mathbb{Z}_p^*$ and h is any collision resistant hash function. The verification algorithm $\mathcal{V}(x, \langle a, b \rangle)$ just checks whether $b = a^{h(x)} \mod p$.

4 Interpretation

Section 2 describes a setting where messages are algebraic terms generated by some grammar. In Section 3 messages are bit strings and operations are given by probabilistic algorithms operating on bit strings. This section shows how to map algebraic messages to (distributions over) bit strings. This interpretation is very much standard. We refer to [AR02,AJ01,MW04] for a thorough explanation. We follow the notation from [GvR06c].

Tagged representation. Throughout this paper we assume that it is always possible to recover the type information of a message from its bit string representation. This can be easily achieved by adding the necessary type tags to the bit string representation. We will abstract from this representation by overloading the notation. We use Greek letters

for bitstrings and μ represents a bit string of a generic type. We write $\mu_1\mu_2$ for a pair of bit strings (in [AR02] this would be written as $\langle(\mu_1,\mu_2),\text{“pair”}\rangle$); ϵ for a ciphertext; κ for a key; ψ for a hash value; ν for a nonce and ς for a constant.

Definition 4.1. For every message m we define the set $\mathrm{R}(m) \subseteq \mathrm{Msg}$ of *random messages in* m as follows:

$$\begin{aligned} \mathrm{R}(c) &= \emptyset & \mathrm{R}(\{\!|m|\!\}_k^r) &= \mathrm{R}(m) \cup \{k, \{\!|m|\!\}_k^r\} \\ \mathrm{R}(n) &= \{n\} & \mathrm{R}(\mathrm{h}^r(m)) &= \mathrm{R}(m) \cup \{\mathrm{h}^r(m)\} \\ \mathrm{R}(k) &= \{k\} & \mathrm{R}(\langle m_1, m_2\rangle) &= \mathrm{R}(m_1) \cup \mathrm{R}(m_2) \\ \mathrm{R}(\square^r) &= \{k_\square, \square^r\} & \mathrm{R}(\boxtimes^r) &= \{n^r_\boxtimes, \boxtimes^r\}. \end{aligned}$$

When interpreting a message m as (ensembles of distributions over) bit strings (Definition 4.3 below), we will first choose a sequence of coin flips for all elements of $\mathrm{R}(m)$ and use these sequences as source of randomness for the appropriate interpretation algorithms.

Definition 4.2. Coins is the set $\{0,1\}^\omega$, the set of all infinite sequences of 0's and 1's. We equip Coins with the probability distribution obtained by flipping a fair coin for each element in the sequence. For every finite set X we define $\mathrm{Coins}(X)$ as $\{\tau\colon X \to \mathrm{Coins}\}$ and equip it with the induced product probability distribution. Furthermore, for every message m we write $\mathrm{Coins}(m)$ instead of $\mathrm{Coins}(\mathrm{R}(m))$.

An element of τ of $\mathrm{Coins}(m)$ gives, for every sub-message m' of m that requires random choices when interpreting this sub-message as a bit string, an infinite sequence $\tau(m')$ of coin flips that will be used to resolve the randomness.

Now we are ready to give semantic to our message algebra. We use $\mathcal{E}$ to interpret encryptions, $\mathcal{K}$ to interpret key symbols, and $\mathcal{H}$ to interpret hashes. We let $\mathcal{C}\colon \mathrm{Const} \to \mathrm{Str}$ be a function that (deterministically) assigns a constant bit string to each constant identifier. We let $\mathcal{N}\colon \mathrm{Param} \to \mathrm{Str}$ be the nonce generation function that, given a unary sequence of length η, chooses uniformly and randomly a bit string from $\{0,1\}^\eta$.

Definition 4.3. For a message m, a value of the security parameter $\eta \in \mathbb{N}$, a finite set U of messages containing $\mathrm{R}(m)$, and for a choice $\tau \in \mathrm{Coins}(U)$ of (at least) all the randomness in m, we can (deterministically) create a bit string $[\![m]\!]^\tau_\eta \in \mathrm{Str}$ as follows:

$$\begin{aligned} [\![c]\!]^\tau_\eta &= \mathcal{C}(c) & [\![\{\!|m|\!\}_k^r]\!]^\tau_\eta &= \mathcal{E}([\![k]\!]^\tau_\eta, [\![m]\!]^\tau_\eta, \tau(\{\!|m|\!\}_k^r)) \\ [\![k]\!]^\tau_\eta &= \mathcal{K}(1^\eta, \tau(k)) & [\![\mathrm{h}^r(m)]\!]^\tau_\eta &= \mathcal{H}(1^\eta, [\![m]\!]^\tau_\eta, \tau(\mathrm{h}^r(m))) \\ [\![n]\!]^\tau_\eta &= \mathcal{N}(1^\eta, \tau(n)) & [\![\square^r]\!]^\tau_\eta &= \mathcal{E}([\![k_\square]\!]^\tau_\eta, \mathcal{C}(0), \tau(\square^r)) \\ [\![\langle m_1, m_2\rangle]\!]^\tau_\eta &= [\![m_1]\!]^\tau_\eta [\![m_2]\!]^\tau_\eta & [\![\boxtimes^r]\!]^\tau_\eta &= \mathcal{H}(1^\eta, [\![n^r_\boxtimes]\!]^\tau_\eta, \tau(\boxtimes^r)). \end{aligned}$$

Note that $[\![m]\!]^\tau_\eta = [\![m]\!]^{\tau|_{\mathrm{R}(m)}}_\eta$. For a fixed message m and $\eta \in \mathbb{N}$, choosing τ from the probability distribution $\mathrm{Coins}(\mathrm{R}(m))$ creates a probability distribution $[\![m]\!]_\eta$ over Str:

$$[\![m]\!]_\eta := [\tau \xleftarrow{\$} \mathrm{Coins}(m); [\![m]\!]^\tau_\eta].$$

Note that although the codomain of $\tau \in \mathrm{Coins}(m)$ is Coins, the set of *infinite* bit strings, when interpreting a fixed message m at a fixed value of the security parameter η, only a predetermined *finite* initial segment of each sequence of coin flips will be used by

$\mathcal{K}$, $\mathcal{N}$, $\mathcal{E}$, and $\mathcal{H}$. Letting η range over $\mathbb{N}$ creates an ensemble of probability distributions $[\![m]\!]$ over Str, namely $[\![m]\!] := \{[\![m]\!]_\eta\}_{\eta\in\mathbb{N}}$. Moreover, we overload the semantic function $[\![-]\!]^\tau_\eta$: Msg $\to$ Str to sets of messages U, in the natural manner:

$$[\![U]\!]^\tau_\eta := \bigcup_{m\in U} [\![m]\!]^\tau_\eta \,.$$

Throughout this paper, when it does not cause confusion, we write $[\![-]\!]^\tau$ instead of $[\![-]\!]^\tau_\eta$ to simplify notation.

In [GvR06c], the following result is proven. Well-spreadness here is a mild condition on the encryption scheme, meaning that no ciphertext is exceptionally likely to occur as the encryption of a particular message. See [GvR06c] for more details. This notion plays no role in the remainder of this paper.

Theorem 4.4 (Soundness) *Assume that the encryption scheme $\langle\mathcal{K},\mathcal{E},\mathcal{D}\rangle$ is type-0 secure and well-spread and that the hash scheme $\langle\mathcal{H},\mathcal{V}\rangle$ is oracle indistinguishable and collision resistant. Let m and m' be acyclic messages. Then $m \cong m' \implies [\![m]\!] \equiv [\![m']\!]$.*

The next section will prove the converse of this under appropriate conditions.

5 Completeness

This section shows that the interpretation proposed in the previous section is complete. Throughout this section we assume that the encryption scheme $\langle\mathcal{K},\mathcal{E},\mathcal{D}\rangle$ is type-0 secure and well-spread and that the probabilistic hash scheme $\langle\mathcal{H},\mathcal{V}\rangle$ is collision resistant and oracle indistinguishable.

Throughout the completeness proof we follow the steps of Micciancio–Warinschi and their notation when possible. We recall here some of their results as there are used in our proof.

In the original Abadi-Rogaway logic, the useful information for an adversary is determined by the set of keys she can learn. We define the function *recoverable* and its computational counterpart *Crecoverable* as in [MW04]. These functions extract the set of keys observable by an adversary from an algebraic message and a bit string respectively.

$$\begin{aligned} \mathit{recoverable}(m) &= \mathit{recoverable}(m, |m|, \emptyset) \\ \mathit{recoverable}(m, d+1, U) &= \mathrm{F}_{\mathrm{kr}}(m, \mathit{recoverable}(m, d, U)) \\ \mathit{recoverable}(m, 0, U) &= \emptyset \end{aligned}$$

$$\begin{aligned} \mathrm{F}_{\mathrm{kr}}(\langle m_1, m_2\rangle, U) &= \mathrm{F}_{\mathrm{kr}}(m_1, U) \cup \mathrm{F}_{\mathrm{kr}}(m_2, U) && \\ \mathrm{F}_{\mathrm{kr}}(k, U) &= \{k\} \cup U && \\ \mathrm{F}_{\mathrm{kr}}(\{\!|m|\!\}^r_k, U) &= \mathrm{F}_{\mathrm{kr}}(m, U), && \text{if } k \in U; \\ \mathrm{F}_{\mathrm{kr}}(m, U) &= U, && \text{in any other case.} \end{aligned}$$

algorithm *Crecoverable*(μ) :
Gets all the keys in the bit string μ with high probability.
$U' := \emptyset$
do:
$U := U'$
$U' := \mathrm{C}_{\mathrm{kr}}(\mu, U)$
until $U = U'$
return U

$$\begin{aligned}
\mathrm{C}_{\mathrm{kr}}(\kappa, U) &= \{\kappa\} \cup U \\
\mathrm{C}_{\mathrm{kr}}(\mu_1\mu_2, U) &= \mathrm{C}_{\mathrm{kr}}(\mu_1, U) \cup \mathrm{C}_{\mathrm{kr}}(\mu_2, U) \\
\mathrm{C}_{\mathrm{kr}}(\epsilon, U) &= \mathrm{C}_{\mathrm{kr}}(\mu, U), \text{if } \exists!\kappa \in U \text{ s.t.} \\
&\qquad\qquad \mathcal{D}(\epsilon, \kappa) = \mu \neq \bot; \\
\mathrm{C}_{\mathrm{kr}}(\mu, U) &= U, \quad \text{otherwise.}
\end{aligned}$$

The following lemma also from [MW04] shows the relation between these two functions.

Lemma 5.1 *Let $\Pi = \langle \mathcal{K}, \mathcal{E}, \mathcal{D} \rangle$ be a confusion free encryption scheme and let $m \in$ Msg. Then*

$$\mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); \mathit{Crecoverable}(\llbracket m \rrbracket^{\tau}) \neq \llbracket \mathit{recoverable}(m) \rrbracket^{\tau}]$$

is a negligible function of η.

Proof. We refer the reader to the original paper for a complete proof of this lemma. The hashes that appear in our logic have no influence at all. □

In our extended logic, due to the hashes, it is not true any more that the only useful information for an adversary are the keys. Any message an adversary can learn might be the pre-image of a certain hash value. Therefore, we need to be able to compute the complete closure of a given message (or bit string). The function *aclosure* below computes the messages in the algebraic closure of a message, up to a certain size. The function *bclosure* is its computational counterpart. Next we show that the proposed functions behave similarly with high probability.

$$\begin{aligned}
\mathit{aclosure}(m, d) &= \mathit{aclosure}(m, d, \mathit{recoverable}(m)) \\
\mathit{aclosure}(m, d, U) &= \mathit{asynth}(\mathit{aanalz}(m, U), d) \\
\mathit{aanalz}(\langle m_1, m_2 \rangle, U) &= \mathit{aanalz}(m_1, U) \cup \mathit{aanalz}(m_2, U) \\
\mathit{aanalz}(\{\!|m|\!\}_k^r, U) &= \{\{\!|m|\!\}_k^r\} \cup \mathit{aanalz}(m, U), && \text{if } k \in U; \\
\mathit{aanalz}(m, U) &= \{m\}, && \text{in any other case.}
\end{aligned}$$

$$\begin{aligned}
\mathit{bclosure}(\mu, d) &= \mathit{bclosure}(\mu, d, \mathit{Crecoverable}(\mu)) \\
\mathit{bclosure}(\mu, d, U) &= \mathit{bsynth}(\mathit{banalz}(\mu, U), d) \\
\mathit{banalz}(\mu_1\mu_2, U) &= \mathit{banalz}(\mu_1, U) \cup \mathit{banalz}(\mu_2, U) \\
\mathit{banalz}(\epsilon, U) &= \{\epsilon\} \cup \mathit{banalz}(\mu, U), && \text{if } \exists!\kappa \in U s.t. \mathcal{D}(\epsilon, \kappa) = \mu \neq \bot; \\
\mathit{banalz}(\mu, U) &= \{\mu\}, && \text{in any other case.}
\end{aligned}$$

algorithm *asynth*(U, d) :
Generates all possible vectors of messages in U of size up to d.
$U_1 = U$
for $i = 2$ to d
  $U_i := \emptyset$
  for each $m \in U$
    for each $v \in U_{i-1}$
      $U_i := U_i \cup \{\langle m, v\rangle\}$
return U_d

algorithm *bsynth*(U, d) :
Generates all possible vectors of bit strings in U of size up to d.
$U_1 = U$
for $i = 2$ to d
  $U_i := \emptyset$
  for each $\mu \in U$
    for each $\omega \in U_{i-1}$
      $U_i := U_i \cup \{\mu\omega\}$
return U_d

Lemma 5.2 *Let* $m \in \mathrm{Msg}, \tau \in \mathrm{Coins}(m)$*, and* $T \subseteq \mathrm{Keys}$*. Then the probability*

$$\mathbb{P}\left[\tau \xleftarrow{\$} \mathrm{Coins}(m); \mathit{banalz}([\![m]\!]^\tau, [\![T]\!]^\tau) \neq [\![\mathit{aanalz}(m, T)]\!]^\tau\right]$$

is a negligible function of η*.*

Proof. The proof follows by induction on the structure of m. The only non-trivial case is $m = \{\!|m_1|\!\}_k^r$.

- If $k \in T$, then $[\![k]\!]^\tau \in [\![T]\!]^\tau$. Next

$$\begin{aligned}
&\mathbb{P}[\mathit{banalz}([\![m]\!]^\tau, [\![T]\!]^\tau) = [\![\{m\} \cup \mathit{aanalz}(m_1, T)]\!]^\tau] \\
&\quad \geq \mathbb{P}[[\![m]\!]^\tau \cup \mathit{banalz}([\![m_1]\!]^\tau, [\![T]\!]^\tau) = [\![\{m\} \cup \mathit{aanalz}(m_1, T)]\!]^\tau \\
&\qquad\qquad \wedge \forall \kappa \in [\![T \setminus k]\!]^\tau : \mathcal{D}([\![m]\!], \kappa) = \bot] \\
&\quad \geq \mathbb{P}[\mathit{banalz}([\![m_1]\!]^\tau, [\![T]\!]^\tau) = [\![\mathit{aanalz}(m_1, T)]\!]^\tau \\
&\qquad\qquad \wedge \forall \kappa \in [\![T \setminus k]\!]^\tau : \mathcal{D}([\![m]\!]^\tau, \kappa) = \bot] \\
&\quad \geq 1 - (\mathbb{P}[\mathit{banalz}([\![m_1]\!]^\tau, [\![T]\!]^\tau) \neq [\![\mathit{aanalz}(m_1, T)]\!]^\tau \\
&\qquad\qquad \vee \exists \kappa \in [\![T \setminus k]\!]^\tau : \mathcal{D}([\![m]\!]^\tau, \kappa) \neq \bot]) \\
&\quad \geq 1 - (\mathbb{P}[\mathit{banalz}([\![m_1]\!]^\tau, [\![T]\!]^\tau) \neq [\![\mathit{aanalz}(m_1, T)]\!]^\tau] \\
&\qquad\qquad + \mathbb{P}[\exists \kappa \in [\![T \setminus k]\!]^\tau : \mathcal{D}([\![m]\!]^\tau, \kappa) \neq \bot]) \\
&\quad \geq 1 - (\varepsilon_1(\eta) + \sum_{\kappa \in [\![T \setminus k]\!]^\tau} \mathbb{P}[\mathcal{D}([\![m]\!]^\tau, \kappa) \neq \bot]) \\
&\quad \geq 1 - (\varepsilon_1(\eta) + \varepsilon_2(\eta) \cdot (|T| - 1)),
\end{aligned}$$

where $\varepsilon_1, \varepsilon_2$ are the negligible functions from the induction hypothesis and confusion freeness respectively.

- If $k \notin T$, then $[\![k]\!]^\tau \notin [\![T]\!]^\tau$. Next

$$\begin{aligned}
\mathbb{P}[\mathit{banalz}([\![m]\!]^\tau, [\![T]\!]^\tau) &= [\![\{m\}]\!]^\tau] \\
&\geq \mathbb{P}\left[[\![m]\!]^\tau = [\![\{m\}]\!]^\tau \wedge \forall \kappa \in [\![T]\!]^\tau : \mathcal{D}([\![m]\!]^\tau, \kappa) = \bot\right] \\
&= 1 - \mathbb{P}\left[\exists \kappa \in [\![T]\!]^\tau : \mathcal{D}([\![m]\!]^\tau, \kappa) = \bot\right] \\
&\geq 1 - \sum_{\kappa \in [\![T]\!]} \mathbb{P}\left[\mathcal{D}([\![m]\!]^\tau, \kappa) = \bot\right] \\
&\geq 1 - \varepsilon(\eta) \cdot |T|,
\end{aligned}$$

where ε is a negligible function due to confusion freeness. □

The following is an extended version of the function *psp* from [MW04], which is the computational counterpart of *pattern*. This function takes a bit string as an argument and tries to recover the pattern associated to it. This means that given as input a sample from $[\![m]\!]$, the function outputs (a renaming of) $pattern(m)$ with high probability. As in [MW04] we let f be an arbitrary (but fixed) injective function that associates an identifier (i.e., an element of $\mathrm{Nonce} \cup \mathrm{Key} \cup \mathrm{Const}$) to each bit string of primitive type (i.e., ν, κ, ς).

$$
\begin{aligned}
psp(\mu_1\mu_2, U) &= \langle psp(\mu_1, U), psp(\mu_2, U)\rangle \\
psp(\epsilon, U) &= \begin{cases} \{\!|psp(\mathcal{D}(\epsilon), U)|\!\}^{\mathcal{R}(\epsilon)}_{f(\kappa)}, & \text{if } \exists!\kappa \in U \text{ s.t. } \mathcal{D}(\epsilon,\kappa) \neq \bot; \\ \square^{\mathcal{R}(\epsilon)}, & \text{otherwise.} \end{cases} \\
psp(\psi, U) &= \begin{cases} \mathrm{h}^{\mathcal{R}(\psi)}(psp(\mu, U)), & \text{if } \exists!\mu \in U \text{ s.t. } \mathcal{V}(\mu,\psi) = 1; \\ \boxtimes^{\mathcal{R}(\psi)}, & \text{otherwise.} \end{cases} \\
psp(\mu, U) &= f(\mu) \qquad \text{in any other case.}
\end{aligned}
$$

Theorem 5.3 *Let $m \in \mathrm{Msg}, \tau \in \mathrm{Coins}(m)$, and U be a finite subset of* Msg. *Then the probability*

$$\mathbb{P}\left[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m]\!]^\tau, [\![U]\!]^\tau) \not\equiv pattern(m, U)\right]$$

is a negligible function of η.

Proof. The proof follows by induction on the structure of m. We only show here the case $m = \mathrm{h}^r(m_1)$. For the remaining cases, the proof follows similarly to the one in the original Micciancio-Warinschi [MW04] paper and therefore we refer the reader to it.

• If $m_1 \in U$ then

$$
\begin{aligned}
&\mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m]\!]^\tau, [\![U]\!]^\tau) \equiv pattern(m, U)] \\
&\geq \mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m_1]\!]^\tau, [\![U]\!]^\tau) \equiv pattern(m_1, U) \\
&\qquad\qquad \wedge \forall \mu \in [\![U \setminus \{m_1\}]\!]^\tau : \mathcal{V}(\mu, [\![m]\!]^\tau) = 0] \\
&= 1 - \mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m_1]\!]^\tau, [\![U]\!]^\tau) \not\equiv pattern(m_1, U) \\
&\qquad\qquad \vee \exists \mu \in [\![U \setminus \{m_1\}]\!]^\tau : \mathcal{V}(\mu, [\![m]\!]^\tau) = 1] \\
&\geq 1 - (\mathbb{P}\left[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m_1]\!]^\tau, [\![U]\!]^\tau) \not\equiv pattern(m_1, U)\right] \\
&\qquad\qquad + \mathbb{P}\left[\tau \xleftarrow{\$} \mathrm{Coins}(m); \exists \mu \in [\![U \setminus \{m_1\}]\!]^\tau : \mathcal{V}(\mu, [\![m]\!]^\tau) = 1\right]) \\
&\geq 1 - (\varepsilon_1(\eta) + \varepsilon_2(\eta)),
\end{aligned}
$$

where $\varepsilon_1, \varepsilon_2$ are the negligible functions from the induction hypothesis and collision resistance respectively.

• If $m_1 \notin U$ then

$$
\begin{aligned}
&\mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m]\!]^\tau, [\![U]\!]^\tau) \equiv pattern(m, U)] \\
&\qquad = \mathbb{P}[psp([\![m]\!]^\tau, [\![U]\!]^\tau) \equiv \boxtimes^r] = \mathbb{P}[\forall \mu \in [\![U]\!]^\tau : \mathcal{V}(\mu, [\![m]\!]^\tau) = 0]
\end{aligned}
$$

therefore

$$\begin{aligned}\mathbb{P}[\tau \xleftarrow{\$} \mathrm{Coins}(m); psp([\![m]\!]^\tau, [\![U]\!]^\tau) \not\equiv pattern(m, U)] \\ = \mathbb{P}[\exists \mu \in [\![U]\!]^\tau : \mathcal{V}(\mu, [\![m]\!]^\tau) = 1] \leq \varepsilon(\eta)\,,\end{aligned}$$

where ε is a negligible function due to collision resistance. □

Lemma 5.4 *Let $m \in \mathrm{Msg}$ and $d \in \mathbb{N}$. Then the probability*

$$\mathbb{P}\left[\mu \xleftarrow{\$} [\![m]\!]; psp(\mu, bclosure(\mu, d)) \not\equiv pattern(m, aclosure(m, d))\right]$$

is a negligible function of η.

Proof.

$$\begin{aligned}&\mathbb{P}\big[\mu \xleftarrow{\$} [\![m]\!]; psp(\mu, bclosure(\mu, d)) \equiv pattern(m, aclosure(m, d))\big] \\ &\geq \mathbb{P}\big[\mu \xleftarrow{\$} [\![m]\!]; psp(\mu, bclosure(\mu, d)) \equiv pattern(m, aclosure(m, d)) \\ &\qquad\qquad \wedge\, bclosure(\mu, d) \equiv aclosure(m, d)\big] \\ &\geq 1 - \Big(\mathbb{P}\big[\mu \xleftarrow{\$} [\![m]\!]; psp(\mu, bclosure(\mu, d)) \not\equiv pattern(m, aclosure(m, d))\big] \\ &\qquad\qquad + \mathbb{P}\big[\mu \xleftarrow{\$} [\![m]\!]; bclosure(\mu, d) \not\equiv aclosure(m, d)\big]\Big) \\ &\geq 1 - (\varepsilon_1(\eta) + \varepsilon_2(\eta))\,,\end{aligned}$$

where $\varepsilon_1, \varepsilon_2$ are negligible functions due to Theorem 5.3 and Lemma 5.2 respectively. □

Theorem 5.5 (Completeness) *Let m_1 and m_2 be acyclic messages. Then $[\![m_1]\!] \equiv [\![m_2]\!] \implies m_1 \cong m_2$.*

Proof. Let us assume that $m_1 \ncong m_2$. Now we show that $[\![m_1]\!] \not\equiv [\![m_2]\!]$ by building a distinguisher D.

algorithm $D(\mu)$:
$d := \max(|m_1|, |m_2|)$
if $psp(\mu, bclosure(\mu, d)) \equiv pattern(m_1)$
    return 1
else
    return 0

Next we show that $\mathrm{Adv}_D(\eta) = |\mathbb{P}[\mu \xleftarrow{\$} [\![m_1]\!]; D(\mu) = 1] - \mathbb{P}[\mu \xleftarrow{\$} [\![m_2]\!]; D(\mu) = 1]|$ is not negligible. On the one hand

$$\begin{aligned}\mathbb{P}[\mu \xleftarrow{\$} [\![m_1]\!]; D(\mu) = 1] &= \mathbb{P}[\mu \xleftarrow{\$} [\![m_1]\!]; psp(\mu, bclosure(\mu, d)) \equiv pattern(m_1)] \\ &= 1 - \mathbb{P}[\mu \xleftarrow{\$} [\![m_1]\!]; psp(\mu, bclosure(\mu, d)) \not\equiv pattern(m_1)] \\ &\geq 1 - \varepsilon_1(\eta)\,,\end{aligned}$$

where ε_1 is the negligible function from Lemma 5.4. Note that $pattern(m_1) = pattern(m_1, aclosure(m_1, |m_1|))$. On the other hand

$$\begin{aligned}\mathbb{P}[\mu \xleftarrow{\$} [\![m_2]\!]; D(\mu) = 1] &= \mathbb{P}[\mu \xleftarrow{\$} [\![m_2]\!]; psp(\mu, bclosure(\mu, d)) \equiv pattern(m_1)] \\ &\leq \mathbb{P}[\mu \xleftarrow{\$} [\![m_2]\!]; psp(\mu, bclosure(\mu, d)) \not\equiv pattern(m_2)] \\ &\leq \varepsilon_2(\eta)\,,\end{aligned}$$

where ε_2 is the negligible function from Lemma 5.4. Therefore, $\mathrm{Adv}_D(\eta) = 1 - \varepsilon_1(\eta) - \varepsilon_2(\eta)$, which is not negligible. □

6 Conclusions

In this paper we have studied an extension of the Abadi-Rogaway logic with hashes. Together with the results from [GvR06c] we have shown that it is possible to create a sound and complete interpretation of formal hashes in the computational world. Under standard assumptions on hashes (pre-image resistance and collision resistance), the algebraic world does not perfectly match the computational world. However, our results show that it is still possible to achieve this perfect match using Canetti's oracle hashing.

References

[AJ01] Martín Abadi and Jan Jrjens. Formal eavesdropping and its computational interpretation. In Naoki Kobayashi and Benjamin C. Pierce, editors, *Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software (TACS'01)*, volume 2215 of *Lecture Notes in Computer Science*, pages 82–94. Springer, 2001.

[AR02] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[BDJR97] Mihir Bellare, Anand Desai, Eron Jokipii, and Philip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science (FOCS'97)*, pages 394–405. IEEE, 1997.

[BPW06] Michael Backes, Birgit Pfitzmann, and Michael Waidner. Limits of the BRSIM/UC soundness of dolev-yao models with hashes. In Dieter Gollmann, Jan Meier, and Andrei Sabelfeld, editors, *Proceedings of the 11th European Symposium on Research in Computer Security (ESORICS 2006)*, volume 4189 of *Lecture Notes in Computer Science*, pages 404–423. Springer, 2006.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burt Kaliski, editor, *Advances in Cryptology, 17th Annual International Cryptology Conference (CRYPTO'97)*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.

[CMR98] Ran Canetti, Danielle Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, pages 131–140. ACM, 1998.

[DY83] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[GB01] Shafi Goldwasser and Mihir Bellare. *Lecture Notes on Cryptography.* 2001. `http://www-cse.ucsd.edu/~mihir/papers/gb.html`.

[Gol01] Oded Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001.

[GvR06a] Flavio D. Garcia and Peter van Rossum. Sound and complete computational interpretation of symbolic hashes in the standard model. In Veronique Cortier and Steve Kremer, editors, *Workshop of Formal and Computational Cryptography (FCC 2006)*, 2006.

[GvR06b] Flavio D. Garcia and Peter van Rossum. Sound computational interpretation of formal hashes. Technical Report ICIS-R06001, Nijmegen Institute for Computing and Information Sciences, `http://www.cs.ru.nl/research/reports/info/ICIS-R06001.html`, 2006.

[GvR06c] Flavio D. Garcia and Peter van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In Hiroshi Yoshiura, Kouichi Sakurai, Kai Rannenberg, Yuko Murayama, and Shinichi Kawamura, editors, *Advances in Information and Computer Security. International Workshop on Security (IWSEC2006)*, volume 4266 of *Lecture Notes in Computer Science*, pages 33–47, Kyoto, Japan, Oct 23-24 2006. Springer Verlag.

[Her05] Jonathan Herzog. A computational interpretation of Dolev-Yao adversaries. *Theoretical Computer Science*, 340(1):57–81, 2005.

[MW04] Daniele Micciancio and Bogdan Warinschi. Completeness theorems of the Abadi-Rogaway logic of encrypted expressions. *Journal of Computer Security*, 12(1):99–129, 2004.

[PW00] Birgit Pfitzmann and Michael Waidner. Composition and integrity preservation of secure reactive systems. In *Proceedings of the 7th ACM CCS*, pages 245–254, 2000.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal Roy and Willi Meier, editors, *Fast Software Encryption: 11th International Workshop (FSE'04)*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.