Computational soundness of equational theories* (Tutorial)

Steve Kremer

LSV, ENS Cachan & CNRS & INRIA Futurs
kremer@lsv.ens-cachan.fr

Abstract. We study the link between formal and cryptographic models for security protocols in the presence of passive and adaptive adversaries. We first describe the seminal result by Abadi and Rogaway and shortly discuss some of its extensions. Then we describe a general model for reasoning about the soundness of implementations of equational theories. We illustrate this model on several examples of computationally sound implementations of equational theories.

1 Introduction

Security protocols have been deployed massively during the last years. However, their security is difficult to ensure and even small protocols are known to be error-prone. Two different approaches for proving such protocols correct have been developed. On the one hand, the *symbolic* or *formal* approach models messages and cryptographic primitives by a term algebra. The adversary manipulates the terms only according to a pre-defined set of rules. On the other hand, the *computational* approach considers a more detailed execution and adversary model. Protocol messages are modelled as bitstrings and cryptographic primitives as algorithms. The adversary is modelled to be any probabilistic polynomial time Turing machine and the security of a protocol is measured as the adversary's success probability.

A considerable advantage of the symbolic model is that proofs can be (at least partially) automated. Unfortunately, it is not clear whether the abstract symbolic model captures all possible attacks. While the computational model provides much stronger security guarantees, proofs are generally harder and difficult to automate. A recent trend tries to get the best of both worlds: an abstract model which provides strong computational guarantees. In their seminal paper, Abadi and Rogaway [4] have shown a first such *soundness result* in the presence of a passive attacker for a simple abstract algebra with symmetric encryption. However, many protocols rely on more complex cryptographic primitives which may have algebraic properties (see [15] for a survey on algebraic properties). Such properties are naturally modelled using equational theories.

In this tutorial paper, we first present the original Abadi and Rogaway result and briefly discuss some of its extensions. Then we present a general framework for reasoning about the soundness of the implementation of an equational theory [10, 19]. The formal indistinguishability relation we consider is static equivalence, a well-established

* Work partly supported by ARA SSIA Formacrypt and ARTIST2 Network of Excellence.

security notion coming from cryptographic pi calculi [3] whose verification can often be automated [2, 11]. A soundness result for an equational theory proves that indeed "enough" equations have been considered in the symbolic model, with respect to a given implementation. We first consider soundness in the presence of a passive adversary and then extend the setting to an adaptive adversary. We present soundness results for several equational theories.

There do also exist soundness results in the presence of an active adversary, notably pioneered by Backes et al. [9] and Micciancio and Warinschi [23]. However, we are not aware of a framework for reasoning about soundness of equational theories with active adversaries which remains a challenging topic of research.

This tutorial is mainly based on joint work with Mathieu Baudet, Véronique Cortier and Laurent Mazaré [10, 19].

2 Preliminaries

Let $f : \mathbb{N} \to \mathbb{R}$ be a function. We say that f is a *negligible* function of η if $f(\eta)$ remains eventually smaller than any η^{-n} $(n > 0)$ for sufficiently large η. Conversely, a function $f(\eta)$ is *overwhelming* if $1 - f(\eta)$ is negligible.

We denote by $\mathcal{A}^{\mathcal{O}}$ the Turing machine $\mathcal{A}$ which has access to the oracles $\mathcal{O}$.

$x \xleftarrow{R} \mathcal{D}$ denotes the random drawing of x from a distribution $\mathcal{D}$.

Let $\eta > 0$ be a complexity parameter and $(\mathcal{D}_\eta)$ a family of distributions, one for each η. A family of distributions $(\mathcal{D}_\eta)$ is *collision-free* iff the probability of collision between two random elements from D_η, that is, $\mathbb{P}[\, e_1, e_2 \xleftarrow{R} \mathcal{D}_\eta \ : \ e_1 = e_2]$, is a negligible function of η.

3 The Abadi-Rogaway result

In this section we summarize the seminal result of Abadi and Rogaway [4, 5]. They show the first soundness result for a simple equivalence on formal expressions. This paper has given raise to many extensions in the passive case and has inspired the generalization to the case of an adaptive and active adversary.

3.1 Formal expressions and equivalence

On the formal side, we consider a simple grammar of formal expressions or terms. The expressions consider two base types for keys and Booleans which are taken from two disjoint sets **Keys** and **Bool**. Keys and Booleans can be paired and encrypted.

$M, N ::=$	$expressions$
K	key ($K \in$ **Keys**)
i	bit ($i \in$ **Bool**)
$\langle M, N \rangle$	pair
$\{M\}_K$	encryption ($K \in$ **Keys**)

For example the formal expression $\langle K_1, \{\langle 0, K_2\rangle\}_{K_1}\rangle$ represents a pair: the first component of this pair is the key K_1, the second, the encryption with key K_1 of the pair consisting of the boolean constant 0 and the key K_2.

Before defining the equivalence relation between terms we first need to define the deducibility relation $\vdash$. Intuitively, $M \vdash N$, if the adversary can learn the expression N from the expression M. Formally, $\vdash$ is the smallest relation, such that

$$\begin{array}{l} M \vdash M \qquad M \vdash 0 \qquad M \vdash 1 \\ \text{if } M \vdash N_1 \text{ and } M \vdash N_2 \text{ then } M \vdash \langle N_1, N_2\rangle \\ \text{if } M \vdash \langle N_1, N_2\rangle \text{ then } M \vdash N_1 \text{ and } M \vdash N_2 \\ \text{if } M \vdash \{N\}_K \text{ and } M \vdash K \text{ then } M \vdash N \\ \text{if } M \vdash N \text{ and } M \vdash K \text{ then } M \vdash \{N\}_K \end{array}$$

For example, if $M = \langle K_1, \{\langle 0, K_2\rangle\}_{K_1}\rangle$, then we have that $M \vdash K_2$. Moreover, $M \vdash 1$, as the constants 0 and 1 are always known to the attacker.

The equivalence relation between terms is based on the equality of the *patterns* associated to each term. A pattern represents the *adversary's view* of a term. Patterns extend the grammar defining terms by the special symbol $\Box$. The pattern of a term replaces encryptions for which the key cannot be deduced by $\Box$. We define the function p, taking as arguments a term and a set T of keys, inductively as follows.

$$\begin{array}{rl} p(K, T) = K & (K \in \textbf{Keys}) \\ p(i, T) = i & (i \in \textbf{Bool}) \\ p(\langle M, N\rangle, T) = \langle p(M, T), p(N, T)\rangle & \\ p(\{M\}_K, T) = \begin{cases} \{p(M, T)\}_K & \text{if } K \in T \\ \Box & \text{else} \end{cases} & \end{array}$$

The pattern of an expression is defined as

$$pattern(M) = p(M, \{K \in \textbf{Keys} \mid M \vdash K\}).$$

For instance $pattern(\langle K_1, \{\langle 0, \{1\}_{K_2}\rangle\}_{K_1}\rangle) = \langle K_1, \{\langle 0, \Box\rangle\}_{K_1}\rangle$.

We say that M and N are formally indistinguishable, written $M \equiv N$ if and only if $pattern(M) = pattern(N)\sigma$, where σ is a bijection on keys (here interpreted as a substitution applied on $pattern(N)$). As an illustration, we have that $0 \not\equiv 1$, $K_0 \equiv K_1$, $\langle K_0, K_0\rangle \not\equiv \langle K_0, K_1\rangle$ and $\{0\}_{K_1} \equiv \{1\}_{K_0}$. Bijective renaming of keys reflects the intuition that two different randomly chosen keys are indistinguishable.

3.2 Computational messages and indistinguishability

In the computational setting, we reason on the level of bitstrings and algorithms executed on Turing Machines, rather than on abstract terms. An encryption scheme in this setting is a triple of polynomial time algorithms $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$, which are the key-generation, encryption and decryption algorithms. The key generation algorithm is parametrized by a security, or complexity parameter $\eta \in 1^*$ and encryption is probabilistic. Intuitively, η defines the key length. As expected we require that $\mathcal{D}_k(\mathcal{E}_k(m, r)) = m$ for any $k \in \mathcal{KG}(\eta)$ and random bitstring r. Moreover, decryption fails and returns $\perp$ in all other cases.

We say that an encryption scheme $\mathcal{SE}$ is type-0 secure, following the terminology of [4], if for any security parameter η and any probabilistic polynomial time Turing machine $\mathcal{A}$ (the adversary) the advantage $\mathrm{Adv}^{\text{type-0}}(\mathcal{A}, \eta, \mathcal{SE}) =$

$$\mathbb{P}\left[k, k' \xleftarrow{R} \mathcal{KG}(\eta) : \mathcal{A}^{\mathcal{E}_k(\cdot), \mathcal{E}_{k'}(\cdot)} = 1\right] - \mathbb{P}\left[k \xleftarrow{R} \mathcal{KG}(\eta) : \mathcal{A}^{\mathcal{E}_k(0), \mathcal{E}_k(0)} = 1\right]$$

is a negligible function of η. By convention, we suppose that adversaries are given access implicitly to as many fresh random coins as needed, as well as the complexity parameter η.

Intuitively, we require that an adversary cannot distinguish the case where he is given two encryption oracles encrypting with two different keys from the case where he is given twice the same encryption oracle always encrypting the constant bitstring representing 0 with the same key. Note that the answers of the second pair of oracles will be distinct each time because encryption is probabilistic. Type-0 security is a message-length and which-key concealing version of the standard semantic security [18]. Message-length concealing means that the encryption hides the length of the plaintext. Which-key concealing means that the fact that two ciphertexts have been encrypted with the same key is hidden.

It is important to note that an encryption scheme respecting the above security definition may be insecure as soon as the adversary is given a *key cycle*. A key cycle is a sequence of keys $K_1, \ldots, K_n$ such that K_{i+1} encrypts (possibly indirectly) K_i and K_n encrypts K_1. An encryption of key K with itself, i.e., $\mathcal{E}_K(K)$ is a key cycle of length 1. An example of a key cycle of size 2 would be $\mathcal{E}_{K_1}(K_2), \mathcal{E}_{K_2}(K_1)$. In Abadi and Rogaway's main result, key cycles are therefore forbidden. This condition can be found in most soundness results[1]. To better understand the problem of key cycles suppose that $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is a semantically secure encryption scheme and let $\mathcal{SE}' = (\mathcal{KG}', \mathcal{E}', \mathcal{D}')$ be defined as follows:

$$\mathcal{KG}' = \mathcal{KG}, \quad \mathcal{E}'_k(m, r) = \begin{cases} \mathcal{E}_k(m, r) \text{ if } m \neq k \\ \mathsf{const} \cdot k \text{ if } m = k \end{cases}, \quad \mathcal{D}'_k(c) = \begin{cases} \mathcal{D}_k(c) & \text{if } c \neq \mathsf{const} \cdot k \\ k & \text{if } c = \mathsf{const} \cdot k \end{cases}$$

where const is a constant such that for any key k, the concatenation $\mathsf{const} \cdot k$ does not belong to the set of possible ciphertexts obtained by $\mathcal{E}$. Obviously, if the attacker is given a key cycle of length 1, *e.g.*, $\mathcal{E}'_k(k, r)$, the attacker directly learns the key. It is also easy to see that $\mathcal{SE}'$ is a semantic secure encryption scheme as it behaves as $\mathcal{SE}$ in nearly all cases (in the security experiment the adversary could make a query for encrypting k with itself only with negligible probability).

The notion of computational indistinguishability requires that an adversary cannot distinguish two (families of) distributions, with better than negligible probability. Let $\mathcal{D} = \{\mathcal{D}_\eta\}$ and $\mathcal{D}' = \{\mathcal{D}'_\eta\}$ be two families of probability distributions, also called *ensembles*. $\mathcal{D}$ and $\mathcal{D}'$ are *compuationally indistinguishable*, written $\mathcal{D} \approx \mathcal{D}'$ if for any η and any probabilistic polynomial time Turing machine $\mathcal{A}$, the advantage

$$\mathrm{Adv}^{\text{IND}}(\mathcal{A}, \eta, \mathcal{D}_\eta, \mathcal{D}'_\eta) = \mathbb{P}\left[x \xleftarrow{R} \mathcal{D}_\eta : \mathcal{A}(x) = 1\right] - \mathbb{P}\left[x \xleftarrow{R} \mathcal{D}'_\eta : \mathcal{A}(x) = 1\right]$$

is a negligible function of η.

[1] A notable exception is [6] where a stronger definition is considered: Key Dependent Message (KDM) security.

3.3 Interpretation of formal expressions and soundness result

To state Abadi and Rogaway's soundness result we have to define an interpretation of formal terms as bitstrings. Bitstrings are tagged using types "key", "bool", "pair" and "ciphertext". The initialize procedure, first draws all the keys using the key generation algorithm $\mathcal{KG}$; $Keys(M)$ denotes the set of keys appearing in the term M. The convert procedure implements encryption using algorithm $\mathcal{E}$.

Initialize$_\eta(M)$
  for $K \in Keys(M)$ do $\tau(K) \xleftarrow{R} \mathcal{KG}(\eta)$

Convert(M)
  if $M = K$ ($K \in$ **Keys**) then
    return ($\tau(K)$,"key")
  if $M = b$ ($b \in$ **Bool**) then
    return b,"bool")
  if $M = \langle M_1, M_2 \rangle$ then
    return (**Convert**(M_1), **Convert**(M_2), "pair")
  if $M = \{M_1\}_K$ then
    $x \xleftarrow{R}$ **Convert**(M_1)
    $y \xleftarrow{R} \mathcal{E}_{\tau(K)}(x)$
    return$(y,$ "ciphertext")

The initialize and convert procedures associate to a formal term M a family of probability distributions $[\![M]\!] = \{[\![M]\!]_\eta\}$, one for each η. Abadi and Rogaway's main result is the following.

Theorem 1. *For any formal expressions M and N that do not contain key cycles, whenever the computational interpretation of the terms uses a type-0 secure encryption scheme, then $M \equiv N$ implies that $[\![M]\!] \approx [\![N]\!]$.*

3.4 Extensions

The above result has known many extensions. We mention some of them here. Laud and Corin [20] allow the use of composed keys. Adão et al. [7] strengthen cryptographic assumptions to allow key cycles. In [8], Adão et al. consider different implementations of encryption allowing which-key and message-length revealing encryption and also consider the case of one-time pad encryption and information-theoretic security. Garcia and van Rossum [17] add (probabilistic) hash functions and Bresson et al. [12] consider modular exponentiation. However, these extensions require to re-define each time a new formal indistinguishability relation extending the classical notion of patterns.

Micciancio and Warinschi [22] also show a *completeness* result: whenever two families of distributions, resulting from the interpretation of two formal terms, are indistinguishable, then the two formal terms are formally inditinguishable. This result requires a stronger security requirement for encryption, which is *authenticated* encryption (see [22] for details). Such a completeness result ensures that no false attacks are reported by the formal model. Adão et al. [8] extend this result to different implementations of encryptions as for soundness.

4 Abstract and computational algebras

To avoid redefining a new model and a new indistinguishability relation for each extension, we define a general model [10, 19] which relies on equational theories and static equivalence.

4.1 Abstract algebras

In the Abadi-Rogaway model symbolic terms were given by a simple grammar modelling encryption with atomic keys, pairs and boolean constants. Here we introduce a more general model—called *abstract algebras*— which consists of term algebras defined over a many-sorted first-order signature and equipped with equational theories.

Specifically, a *signature* $(\mathcal{S}, \mathcal{F})$ is made of a set of *sorts* $\mathcal{S} = \{s, s_1 \ldots\}$ and a set of *symbols* $\mathcal{F} = \{f, f_1 \ldots\}$ together with arities of the form $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$, $k \geq 0$. Symbols that take $k = 0$ arguments are called *constants*; their arity is simply written s. We fix a set of *names* $\mathcal{N} = \{a, b \ldots\}$ and a set of *variables* $\mathcal{X} = \{x, y \ldots\}$. We assume that names and variables are given with sorts. By default, we assume that an infinite number of names and variables are available for each sort. The set of *terms of sort* s is defined inductively by

$t ::=$		term of sort s
$\mid$	x	variable x of sort s
$\mid$	a	name a of sort s
$\mid$	$f(t_1, \ldots, t_k)$	application of symbol $f \in \mathcal{F}$

where for the last case, we further require that t_i is a term of some sort s_i and $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$. We also allow subsorts: if s_2 is a subsort of s_1 we allow a term of sort s_2 whenever a term of sort s_1 is expected. We write $\mathrm{var}(t)$ and $\mathrm{names}(t)$ for the set of variables and names occurring in t, respectively. A term t is *ground* or *closed* iff $\mathrm{var}(t) = \emptyset$.

Substitutions are written $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ with domain $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$. We only consider *well-sorted*, *cycle-free* substitutions. Such a σ is *closed* iff all of the t_i are closed. We let $\mathrm{var}(\sigma) = \bigcup_i \mathrm{var}(t_i)$, $\mathrm{names}(\sigma) = \bigcup_i \mathrm{names}(t_i)$, and extend the notations $\mathrm{var}(.)$ and $\mathrm{names}(.)$ to tuples and sets of terms and substitutions in the obvious way. The application of a substitution σ to a term t is written $\sigma(t) = t\sigma$ and is defined in the usual way.

Symbols in $\mathcal{F}$ are intended to model cryptographic primitives, whereas names in $\mathcal{N}$ are used to model secrets, that is, for example random numbers or keys. The abstract semantics of symbols is described by an equational theory E, *i.e*, an equivalence relation (also written $=_E$) which is stable by application of contexts and well-sorted substitutions of variables. For instance, symmetric encryption is modeled by the theory E_{enc} generated by the equation $E_{\mathsf{enc}} = \{\mathrm{dec}(\mathrm{enc}(x, y), y) = x\}$.

4.2 Deducibility and static equivalence

We use frames [3, 2] to represent sequences of messages observed by an attacker, for instance during the execution of a protocol. Formally, a *frame* is an expression $\varphi =$

$\nu\tilde{a}.\{x_1 = t_1, \ldots, x_n = t_n\}$ where $\tilde{a}$ is a set of *bound (or restricted) names*, and for each i, t_i is a closed term of the same sort as x_i.

For simplicity, we only consider frames $\varphi = \nu\tilde{a}.\{x_1 = t_1, \ldots, x_n = t_n\}$ which restrict every name in use, that is $\tilde{a} = \mathrm{names}(t_1, \ldots, t_n)$. A name a may still be disclosed explicitly by adding a mapping $x_a = a$ to the frame. Thus we tend to assimilate such frames φ to their *underlying substitutions* $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$.

In the previous section, we introduced deducibility and formal indistinguishability for the simple term algebra of encryption and pairing. We now define similar notions with respect to an equational theory.

Definition 1 (Deducibility). *A (closed) term t is* deducible *from a frame φ in an equational theory E, written $\varphi \vdash_E t$, iff there exists a term M such that* $\mathrm{var}(M) \subseteq \mathrm{dom}(\varphi)$, $\mathrm{names}(M) \cap \mathrm{names}(\varphi) = \emptyset$, *and* $M\varphi =_E t$.

In what follows, again for simplicity, we only consider deducibility problems $\varphi \vdash_E t$ such that $\mathrm{names}(t) \subseteq \mathrm{names}(\varphi)$. Consider for instance the theory E_{enc} and the frame $\varphi_1 = \{x_1 \mapsto \mathrm{enc}(k_1, k_2),\ x_2 \mapsto \mathrm{enc}(k_4, k_3),\ x_3 \mapsto k_3\}$: the name k_4 is deducible from φ_1 since $\mathrm{dec}(x_2, x_3)\varphi_1 =_{E_{\mathsf{enc}}} k_4$ but neither are k_1 nor k_2. Deducibility is not always sufficient to account for the knowledge of an attacker. For instance, it lacks partial information on secrets. We refer the reader to [2] for additional details and examples. That is why another classical notion in formal methods is *static equivalence*, which will be our formal indistinguishability relation.

Definition 2 (Static equivalence). *Two frames φ_1 and φ_2 are* statically equivalent *in a theory E, written $\varphi_1 \approx_E \varphi_2$, iff* $\mathrm{dom}(\varphi_1) = \mathrm{dom}(\varphi_2)$, *and for all terms M and N such that* $\mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi_1)$ *and* $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi_1, \varphi_2) = \emptyset$, $M\varphi_1 =_E N\varphi_1$ *is equivalent to* $M\varphi_2 =_E N\varphi_2$.

For instance, consider the equational theory E_{enc} of symmetric encryption. Let 0 and 1 be two constants (which are thus known by the attacker). Then the two frames $\{x \mapsto \mathrm{enc}(0, k)\}$ and $\{x \mapsto \mathrm{enc}(1, k)\}$ are statically equivalent with respect to E_{enc}. However $\varphi = \{x \mapsto \mathrm{enc}(0, k),\ y \mapsto k\}$ and $\varphi' = \{x \mapsto \mathrm{enc}(1, k),\ y \mapsto k\}$ are not statically equivalent for E_{enc}: let M be the term $\mathrm{dec}(x, y)$ and N be the term 0. M and N use only variables defined by φ and φ' and do not use any names. Moreover $M\varphi =_{E_{\mathsf{enc}}} N\varphi$ but $M\varphi' \neq_{E_{\mathsf{enc}}} N\varphi'$. The test $M \stackrel{?}{=} N$ distinguishes φ from φ'.

4.3 Concrete semantics

We now give terms and frames a concrete semantics, parameterized by an implementation of the primitives. Provided a set of sorts $\mathcal{S}$ and a set of symbols $\mathcal{F}$ as above, a *$(\mathcal{S}, \mathcal{F})$-computational algebra A* consists of

- a non-empty set of bitstrings $[\![s]\!]_A \subseteq \{0,1\}^*$ for each sort $s \in \mathcal{S}$; moreover, if s_2 is a subsort of s_1 we require that $[\![s_2]\!]_A \subseteq [\![s_1]\!]_A$;
- a computable function $[\![f]\!]_A : [\![s_1]\!]_A \times \ldots \times [\![s_k]\!]_A \to [\![s]\!]_A$ for each $f \in \mathcal{F}$ with $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$;

– a computable congruence $=_{A,s}$ for each sort s, in order to check the equality of elements in $[\![s]\!]_A$ (the same element may be represented by different bitstrings); by congruence, we mean a reflexive, symmetric, transitive relation such that $e_1 =_{A,s_1} e'_1, \ldots, e_k =_{A,s_k} e'_k \Rightarrow [\![f]\!]_A(e_1, \ldots, e_k) =_{A,s} [\![f]\!]_A(e'_1, \ldots, e'_k)$ (in the remaining we often omit s and write $=_A$ for $=_{A,s}$);
– an effective procedure to draw random elements from $[\![s]\!]_A$; we denote such a drawing by $x \xleftarrow{R} [\![s]\!]_A$.

Assume a fixed $(\mathcal{S}, \mathcal{F})$-computational algebra A. We associate to each frame $\varphi = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ a distribution $\psi = [\![\varphi]\!]_A$, of which the drawings $\widehat{\psi} \xleftarrow{R} \psi$ are computed as follows:

1. for each name a of sort s appearing in $t_1, \ldots, t_n$, draw a value $\widehat{a} \xleftarrow{R} [\![s]\!]_A$;
2. for each x_i $(1 \leq i \leq n)$ of sort s_i, compute $\widehat{t_i} \in [\![s_i]\!]_A$ recursively on the structure of terms: $\widehat{f(t'_1, \ldots, t'_m)} = [\![f]\!]_A(\widehat{t'_1}, \ldots, \widehat{t'_m})$;
3. return the value $\widehat{\psi} = \{x_1 \mapsto \widehat{t_1}, \ldots, x_n \mapsto \widehat{t_n}\}$.

Such values $\phi = \{x_1 = e_1, \ldots, x_n = e_n\}$ with $e_i \in [\![s_i]\!]_A$ are called *concrete frames*. We extend the notation $[\![.]\!]_A$ to (tuples of) closed terms in the obvious way.

We focus on asymptotic notions of cryptographic security and consider families of computational algebra (A_η) indexed by a complexity parameter $\eta > 0$. As in previous section, the *concrete semantics* of a frame φ is a family of distributions over concrete frames $([\![\varphi]\!]_{A_\eta})$. We only consider families of computational algebras (A_η) such that each required operation on algebras is feasible by a (uniform, probabilistic) polynomial-time algorithm in the complexity parameter η. This ensures that the concrete semantics of terms and frames is efficiently computable (in the same sense).

5 Relating abstract and computational algebras

In the previous section we have defined abstract and computational algebras. We now relate formal notions such as equality, (non-)deducibility and static equivalence to their computational counterparts, that is, equality, one-wayness and indistinguishability.

5.1 Soundness and faithfulness

We introduce the notions of sound and faithful computational algebras with respect to the formal relations studied here: equality, static equivalence and deducibility.

Let E be an equational theory. A family of computational algebras (A_η) is

– $=_E$*-sound* iff for every closed terms T_1, T_2 of the same sort, $T_1 =_E T_2$ implies that $\mathbb{P}[\, e_1, e_2 \xleftarrow{R} [\![T_1,\, T_2]\!]_{A_\eta} \;:\; e_1 =_{A_\eta} e_2]$ is overwhelming;
– $=_E$*-faithful* iff for every closed terms T_1, T_2 of the same sort, $T_1 \neq_E T_2$ implies that $\mathbb{P}[\, e_1, e_2 \xleftarrow{R} [\![T_1,\, T_2]\!]_{A_\eta} \;:\; e_1 =_{A_\eta} e_2]$ is negligible;
– $\approx_E$*-sound* iff for every frames φ_1, φ_2 with the same domain, $\varphi_1 \approx_E \varphi_2$ implies that $([\![\varphi_1]\!]_{A_\eta}) \approx ([\![\varphi_2]\!]_{A_\eta})$;

- $\approx_E$*-faithful* iff for every frames φ_1, φ_2 of the same domain, $\varphi_1 \not\approx_E \varphi_2$ implies that there exists a polynomial-time adversary $\mathcal{A}$ for distinguishing concrete frames, such that $\mathrm{Adv}^{\mathrm{IND}}(\mathcal{A}, \eta, [\![\varphi_1]\!]_{A_\eta}, [\![\varphi_2]\!]_{A_\eta})$ is overwhelming;
- $\not\vdash_E$*-sound* iff for every frame φ and closed term T such that $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi)$, $\varphi \not\vdash_E T$ implies that for each polynomial-time adversary $\mathcal{A}$, we have that the probability $\mathbb{P}[\phi, e \xleftarrow{R} [\![\varphi, T]\!]_{A_\eta} : \mathcal{A}(\phi) =_{A_\eta} e]$ is negligible;
- $\not\vdash_E$*-faithful* iff for every frame φ and closed term T such that $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi)$, $\varphi \vdash_E T$ implies that there exists a polynomial-time adversary $\mathcal{A}$ such that the probability $\mathbb{P}[\phi, e \xleftarrow{R} [\![\varphi, T]\!]_{A_\eta} : \mathcal{A}(\phi) =_{A_\eta} e]$ is overwhelming.

We note that faithfullness is stronger than completeness as defined in [22]. It requires that whenever static equivalence does not hold distributions can be distinguished *efficiently*. Completeness could be defined by replacing "overwhelming" with "non-negligible". Sometimes, it is possible to prove stronger notions of soundness that hold without restriction on the computational power of adversaries. In particular, (A_η) is

- *unconditionally* $=_E$*-sound* iff for every closed terms T_1, T_2 of the same sort, $T_1 =_E T_2$ implies that $\mathbb{P}[\, e_1, e_2 \xleftarrow{R} [\![T_1,\, T_2]\!]_{A_\eta} : e_1 =_{A_\eta} e_2] = 1$;
- *unconditionally* $\approx_E$*-sound* iff for every frames φ_1, φ_2 with the same domain, $\varphi_1 \approx_E \varphi_2$ implies $([\![\varphi_1]\!]_{A_\eta}) = ([\![\varphi_2]\!]_{A_\eta})$;
- *unconditionally* $\not\vdash_E$*-sound* iff for every frame φ and closed term T such that $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi)$ and $\varphi \not\vdash_E T$, the drawings for φ and T are independent: for all ϕ_0, e_0, $\mathbb{P}[\phi_0, e_0 \xleftarrow{R} [\![\varphi, T]\!]_{A_\eta}] = \mathbb{P}[\phi_0 \xleftarrow{R} [\![\varphi]\!]_{A_\eta}] \times \mathbb{P}[e_0 \xleftarrow{R} [\![T]\!]_{A_\eta}]$, and the drawing $(\xleftarrow{R} [\![T]\!]_{A_\eta})$ is collision-free.

Generally, (unconditional) $=_E$-soundness is given by construction. Indeed true formal equations correspond to the expected behavior of primitives and should hold in the concrete world with overwhelming probability. The other criteria are however more difficult to fulfill. Therefore it is often interesting to restrict frames to *well-formed* ones in order to achieve soundness or faithfulness: we have already encountered a typical example of such a restriction which was to forbid key cycles.

It is worth noting that the notions of soundness and faithfulness introduced above are not independent.

Proposition 1. *Let* (A_η) *be a* $=_E$*-sound family of computational algebras. Then*

1. (A_η) *is* $\not\vdash_E$*-faithful;*
2. *if* (A_η) *is also* $=_E$*-faithful,* (A_η) *is* $\approx_E$*-faithful.*

For many theories, we have that $\approx_E$-soundness implies all the other notions of soundness and faithfulness. This emphasizes the importance of $\approx_E$-soundness and provides an additional motivation for its study. As an illustration, let us consider an arbitrary theory which includes keyed hash functions.

A symbol f is *free* with respect to an equational theory E iff there exists a set of equations F generating E such that f does not occur in F. A sort s is *degenerated* in E iff all terms of sort s are equal modulo E.

Proposition 2. *Let (A_η) be a family of $\approx_E$-sound computational algebras. Assume that free binary symbols $\mathsf{h}_s : s \times Key \to Hash$ are available for every sort s, where the sort Key is not degenerated in E, and the drawing of random elements for the sort $Hash$, $(\xleftarrow{R} [\![Hash]\!]_{A_\eta})$, is collision-free. Then*

1. *(A_η) is $=_E$-faithful;*
2. *(A_η) is $\not\models_E$-sound;*
3. *Assume the implementations for the h_s* collision-resistant *in the sense that for all T_1, T_2 of sort s, given a fresh name k of sort Key, the quantity*

$$\mathbb{P}\left[e_1, e_2, e_1', e_2' \xleftarrow{R} [\![T_1, T_2, \mathsf{h}_s(T_1, k), \mathsf{h}_s(T_2, k)]\!]_{A_\eta} \ : \ e_1 \neq_{A_\eta} e_2,\ e_1' =_{A_\eta} e_2'\right]$$

is negligible. Then (A_η) is $=_E$-sound, $\not\models_E$-faithful and $\approx_E$-faithful.

6 Examples

We now illustrate the framework by several examples. Details and proofs can be found in [10, 19].

6.1 Exclusive OR

We study the soundness and faithfulness problems for the natural theory and implementation of the exclusive OR (XOR), together with constants and (pure) random numbers.

The formal model consists of a single sort $Data_\oplus$, an infinite number of names, the infix symbol $\oplus : Data_\oplus \times Data_\oplus \to Data_\oplus$ and two constants $0, 1 : Data_\oplus$. Terms are equipped with the equational theory $E_\oplus$ generated by:

$$(x \oplus y) \oplus z = x \oplus (y \oplus z) \qquad x \oplus y = y \oplus x \qquad x \oplus x = 0 \qquad x \oplus 0 = x$$

As an implementation, we define the computational algebras A_η, $\eta \geq 0$:

- the concrete domain $[\![Data_\oplus]\!]_{A_\eta}$ is the set of bitstrings of length η, $\{0,1\}^\eta$, equipped with the uniform distribution;
- $\oplus$ is interpreted by the usual XOR function over $\{0,1\}^\eta$;
- $[\![0]\!]_{A_\eta} = 0^\eta$ and $[\![1]\!]_{A_\eta} = 1^\eta$.

Theorem 2. *The implementation of XOR for the considered signature, (A_η), is unconditionally $=_{E_\oplus}$-, $\approx_{E_\oplus}$- and $\not\models_{E_\oplus}$-sound. It is also $=_{E_\oplus}$-, $\approx_{E_\oplus}$- and $\not\models_{E_\oplus}$-faithful.*

6.2 Modular exponentiation

As another application, we study soundness of modular exponentiation. The cryptographic assumption we make is that the *Decisional Diffie-Hellman* (DDH) problem is difficult: even when given g^x and g^y, it is difficult for any feasible computation to distinguish between g^{xy} and g^r, when x, y and r are selected at random. The original Diffie-Hellman protocol has been used as a building block for several key agreement protocols that are widely used in practice (e.g. SSL/TLS and Kerberos V5).

Symbolic model. The symbolic model consists of two sorts G (group elements) and R (ring elements), an infinite number of names for R, no name for sort G and the symbols:

$$\begin{array}{rlrl} +, \cdot : R \times R \to R & \text{add, mult} & \exp : R \to G & \text{exponentiation} \\ - : R \to R & \text{inverse} & * : G \times G \to G & \text{mult in } \mathbb{G} \\ 0_R, 1_R : R & \text{constants} & & \end{array}$$

We consider the equational theory E_{DH} generated by:

$$\begin{array}{ccc} x + y = y + x & x \cdot y = y \cdot x & (x + y) + z = x + (y + z) \\ x \cdot (y + z) = x \cdot y + x \cdot z & (x \cdot y) \cdot z = x \cdot (y \cdot z) & x + (-x) = 0_R \\ 0_R + x = x & 1_R \cdot x = x & \exp(x) * \exp(y) = \exp(x + y) \end{array}$$

There exists a direct correspondence between terms of sort R and the set of polynomials $\mathbb{Z}[\mathcal{N}_R]$ where $\mathcal{N}_R$ is the set of names of sort R. An integer i simply corresponds to $\underbrace{1_R + \ldots + 1_R}_{i \text{ times}}$ if $i > 0$, to $-(\underbrace{1_R + \ldots + 1_R}_{i \text{ times}})$ if $i < 0$ and to 0_R if $i = 0$. We also write x^n for $\underbrace{x \cdot \ldots \cdot x}_{n \text{ times}}$. This correspondence can be exploited to decide static equivalence [19].

We put two restrictions on formal terms: products have to be *power-free*, *i.e.*, x^n is forbidden for $n > 1$, and products must not contain more than l elements for some fixed bound l, i.e. $x_1 \cdot \ldots \cdot x_n$ is forbidden for $n > l$. Both restrictions come from the DDH assumption and seem difficult to avoid [12]. Furthermore we are only interested in frames using terms of sort G.

Concrete model. An Instance Generator IG is a polynomial-time (in η) algorithm that outputs a cyclic group $\mathbb{G}$ (defined by a generator g, an order q and a polynomial-time multiplication algorithm) of prime order q. The family of computational algebras (A_η) depends on an instance generator IG that generates a cyclic group $\mathbb{G}$ of generator g and of order q: the concrete domain $[\![R]\!]_{A_\eta}$ is $\mathbb{Z}_q$ with the uniform distribution. Symbols $+$ and $\cdot$ are the classical addition and multiplication over $\mathbb{Z}_q$, exp is interpreted as modular exponentiation of g. Constants 0_R and 1_R are respectively interpreted by integers 0 and 1 of $\mathbb{Z}_q$. The domain $[\![G]\!]_{A_\eta}$ contains all the bitstrings representation of elements of $\mathbb{G}$.

A family of computational algebras satisfies the DDH assumption if its instance generator satisfies the assumption, *i.e.* for every probabilistic polynomial-time adversary $\mathcal{A}$, we have that his advantage $\mathcal{A}$, $\mathrm{Adv}^{\mathsf{DDH}}(\mathcal{A}, \eta, IG)$, defined as

$$\begin{array}{l} \mathbb{P}\left[(g, q) \leftarrow IG(\eta) \ : \ a, b \leftarrow \mathbb{Z}_q \ : \ \mathcal{A}(g^a, g^b, g^{ab}) = 1\right] - \\ \mathbb{P}\left[(g, q) \leftarrow IG(\eta) \ : \ a, b, c \leftarrow \mathbb{Z}_q \ : \ \mathcal{A}(g^a, g^b, g^c) = 1\right] \end{array}$$

is negligible in η. We suppose that for any η there is a unique group given by IG. We show that the DDH assumption is necessary and sufficient to prove soundness of $\approx_{E_{\mathsf{DH}}}$.

Theorem 3. *Let (A_η) be a family of computational algebras. (A_η) is $\approx_{E_{\mathsf{DH}}}$-sound iff (A_η) satisfies the* DDH *assumption.*

6.3 Ciphers and lists

We now detail the example of symmetric, deterministic and length-preserving encryption schemes. Such schemes, also known as *ciphers* [24], are widely used in practice, the most famous examples being DES and AES.

Symbolic model. Our formal model consists of a set of sorts $\mathcal{S} = \{Data, List_0, List_1 \dots List_n \dots\}$, an infinite number of names for every sort $Data$ and $List_n$, and the following symbols (for every $n \geq 0$):

$$\begin{array}{rll}
\mathrm{enc}_n, \mathrm{dec}_n : & List_n \times Data \to List_n & \text{encryption, decryption} \\
\mathsf{cons}_n : & Data \times List_n \to List_{n+1} & \text{list constructor} \\
\mathsf{head}_n : & List_{n+1} \to Data & \text{head of a list} \\
\mathsf{tail}_n : & List_{n+1} \to List_n & \text{tail of a list} \\
\mathsf{nil} : & List_0 & \text{empty list} \\
0, 1 : & Data & \text{constants}
\end{array}$$

We consider the equational theory E_{cipher} generated by the following equations (for every $n \geq 0$ and for every name a_0 of sort $List_0$):

$$\begin{array}{rr}
\mathrm{dec}_n(\mathrm{enc}_n(x,y),y) = x & \mathrm{enc}_0(\mathsf{nil},x) = \mathsf{nil} \\
\mathrm{enc}_n(\mathrm{dec}_n(x,y),y) = x & \mathrm{dec}_0(\mathsf{nil},x) = \mathsf{nil} \\
\mathsf{head}_n(\mathsf{cons}_n(x,y)) = x & \mathsf{tail}_0(x) = \mathsf{nil} \\
\mathsf{tail}_n(\mathsf{cons}_n(x,y)) = y & a_0 = \mathsf{nil} \\
\mathsf{cons}_n(\mathsf{head}_n(x), \mathsf{tail}_n(x)) = x &
\end{array}$$

where x, y are variables of the appropriate sorts. The effect of the last four equations is that the sort $List_0$ is degenerated in E_{cipher} (all terms of sort $List_0$ are equal).

Notice that each well-sorted term has a unique sort. As the subscripts n of function symbols are redundant with sorts, we tend to omit them in terms. For instance, if $k, k' : Data$, we may write $\mathrm{enc}(\mathsf{cons}(k, \mathsf{nil}), k')$ instead of $\mathrm{enc}_1(\mathsf{cons}_0(k, \mathsf{nil}), k')$.

The concrete meaning of sorts and symbols is given by the computational algebras A_η, $\eta > 0$, defined as follows:

- the carrier sets are $[\![Data]\!]_{A_\eta} = \{0,1\}^\eta$ and $[\![List_n]\!]_{A_\eta} = \{0,1\}^{n\eta}$ equipped with the uniform distribution and the usual equality relation;
- $\mathrm{enc}_n, \mathrm{dec}_n$ are implemented by a cipher for data of size $n\eta$ and keys of size η (we discuss the required cryptographic assumptions later). Since they are length-preserving they verify the equation $\mathrm{enc}_n(\mathrm{dec}_n(x,y),y) = x$;
- $[\![\mathsf{nil}]\!]_{A_\eta}$ is the empty bitstring, $[\![\mathsf{cons}_n]\!]_{A_\eta}$ is the usual concatenation, $[\![0]\!]_{A_\eta} = 0^\eta$, $[\![1]\!]_{A_\eta} = 1^\eta$, $[\![\mathsf{head}_n]\!]_{A_\eta}$ returns the η first digits of bitstrings (of size $(n+1)\eta$) whereas $[\![\mathsf{tail}_n]\!]_{A_\eta}$ returns the last $n\eta$ digits.

For simplicity we assume without loss of generality that encryption keys have the same size η as blocks of data. We also assume that keys are generated according to the uniform distribution. It is not difficult to prove that the above implementation is unconditionally $=_{E_{\mathsf{cipher}}}$-sound.

Concrete model. We now study the $\approx_{E_{\mathsf{cipher}}}$-soundness problem under classical cryptographic assumptions. Standard assumptions on ciphers include the notions of super pseudo-random permutation (SPRP) and several notions of indistinguishability. In particular, IND-P1-C1 denotes the indistinguishability against lunchtime chosen-plaintext and chosen-ciphertext attacks. These notions and the relations between them have been studied notably in [24].

Initially, the SPRP and IND-P1-C1 assumptions apply to (block) ciphers specialized to plaintexts of a given size. Interestingly, this is not sufficient to imply $\approx_{E_{\mathsf{cipher}}}$-soundness for frames which contain plaintexts of heterogeneous sizes, encrypted under the same key. Thus we introduce a strengthened version of IND-P1-C1, applying to a *collection* of ciphers $(\mathcal{E}_{\eta,n}, \mathcal{D}_{\eta,n})$, where η is the complexity parameter and $n \geq 0$ is the number of blocks of size η contained in plaintexts and ciphertexts.

We define the ω-IND-P1-C1 assumption by considering the following experiment $\mathcal{G}_\eta$ with a 2-stage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

- first a key k is randomly chosen from $\{0,1\}^\eta$;
- (Stage 1) $\mathcal{A}_1$ is given access to the encryption oracles $\mathcal{E}_{\eta,n}(\cdot, k)$ and the decryption oracles $\mathcal{D}_{\eta,n}(\cdot, k)$; it outputs two plaintexts $m_0, m_1 \in \{0,1\}^{n_0\eta}$ for some n_0, and possibly some data d;
- (Stage 2) a random bit $b \in \{0,1\}$ is drawn; $\mathcal{A}_2$ receives the data d, the *challenge ciphertext* $c = \mathcal{E}_{\eta,n_0}(m_b, k)$ and outputs a bit b';
- $\mathcal{A}$ *is successful in* $\mathcal{G}_\eta$ iff $b = b'$ and it has never submitted m_0 or m_1 to an encryption oracle, nor c to a decryption oracle.

Define the *advantage* of $\mathcal{A}$ as

$$\mathrm{Adv}^{\omega\text{-IND-P1-C1}}(\mathcal{A}, \eta) = 2 \times \mathbb{P}\left[\mathcal{A} \text{ is successful in } \mathcal{G}_\eta\right] - 1 \tag{1}$$

The ω-IND-P1-C1 assumption holds for $(\mathcal{E}_{\eta,n}, \mathcal{D}_{\eta,n})$ iff the advantage of any probabilistic polynomial-time adversary is negligible. It holds for the *inverse* of the encryption scheme iff it holds for the collection of ciphers $(\mathcal{D}_{\eta,n}, \mathcal{E}_{\eta,n})$.

We now state our $\approx_{E_{\mathsf{cipher}}}$-soundness theorem. To define well-formed frames we orient the equations of E_{cipher} from left to right which forms a convergent rewriting system $\mathcal{R}$. A closed frame is *well-formed* iff its $\mathcal{R}$-normal form has only atomic keys, contains no encryption cycles and uses no head and tail symbols.

Theorem 4 ($\approx_{E_{\mathsf{cipher}}}$-soundness). *Let φ_1 and φ_2 be two well-formed frames of the same domain. Assume that the concrete implementations for the encryption and its inverse satisfy both the ω-IND-P1-C1 assumption. If $\varphi_1 \approx_{E_{\mathsf{cipher}}} \varphi_2$ then $(\llbracket\varphi_1\rrbracket_{A_\eta}) \approx (\llbracket\varphi_2\rrbracket_{A_\eta})$.*

Cryptographic assumptions of Theorem 4 may appear strong compared to existing work on passive adversaries [4, 22]. This seems unavoidable when we allow frames to contain both encryption and decryption symbols.

6.4 A theory for guessing attacks

In the context of password based protocols and guessing attacks, Abadi et al. [1] consider a complex equational theory: it accounts for symmetric and asymmetric encryption, as well as ciphers that can use passwords as keys. Security against guessing attacks

can be elegantly modelled using static equivalence [14]. The main result is soundness of static equivalence for this equational theory. A direct corollary is soundness of security against guessing attacks. Because of lack of space we will not detail this result.

7 Adaptive soundness

In [19], we extend soundness of static equivalence to the adaptive setting from [21]. In $\approx_E$-soundness the adversary observes the computational value of a fixed frame whereas in this setting the adversary sees the computational value of a sequence of adaptively chosen frames. Applications of this adaptive setting include the analysis of multicast key distribution protocols [21] and dynamic group key exchange protocols [19].

The adaptive setting is formalized through a cryptographic game. Let (A_η) be a family of computational algebras and $\mathcal{A}$ be an adversary. $\mathcal{A}$ has access to a left-right evaluation oracle $\mathcal{O}_{LR}$: given a pair of terms (t_0, t_1) it outputs either the implementation of t_0 or t_1. This oracle depends on a selection bit b and uses a local store to record values generated for the different names (these values are used when processing further queries). With a slight abuse of notation, we omit this store and write:

$$\mathcal{O}^b_{LR,A_\eta}(t_0, t_1) = [\![t_b]\!]_{A_\eta}$$

Adversary $\mathcal{A}$ plays an indistinguishability game and its objective is to find the value of b. Formally the advantage of $\mathcal{A}$ is defined by:

$$\mathrm{Adv}^{\mathsf{ADPT}}(\mathcal{A}, \eta, A_\eta) = \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR,A_\eta}} = 1\right] - \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR,A_\eta}} = 1\right]$$

Without further restrictions on the queries made by the adversary, having a non-negligible advantage is easy in most cases. For example the adversary could submit a pair $(0, 1)$ to his oracle. We therefore require the adversary to be *legal*.

Definition 3 (Adaptive soundness). *An adversary $\mathcal{A}$ is* legal *if for any sequence of queries $(t^i_0, t^i_1)_{1\leq i\leq n}$ made by $\mathcal{A}$ to its left-right oracle, queries are statically equivalent:*

$$\left\{x_1 \mapsto t^1_0, \ldots, x_n \mapsto t^n_0\right\} \approx_E \left\{x_1 \mapsto t^1_1, \ldots, x_n \mapsto t^n_1\right\}$$

A family of computational algebras (A_η) is

- $\approx_E$-ad-sound *iff the advantage $\mathrm{Adv}^{\mathsf{ADPT}}(\mathcal{A}, \eta, A_\eta)$ of any polynomial-time legal adversary $\mathcal{A}$ is negligible.*
- unconditionally $\approx_E$-ad-sound *iff the advantage $\mathrm{Adv}^{\mathsf{ADPT}}(\mathcal{A}, \eta, A_\eta)$ of any legal adversary $\mathcal{A}$ is 0.*

Note that as variables are typed, any query (t^i_0, t^i_1) of a legal adversary to the oracle is such that t^i_0 and t^i_1 have the same sort. Adaptive soundness implies the original soundness notion for static equivalence.

Proposition 3. *Let (A_η) be a family of computational algebras. If A_η is $\approx_E$-ad-sound then A_η is also $\approx_E$-sound but the converse is false in general.*

Interestingly, in the case of unconditional soundness, adaptive and non-adaptive soundness coincide.

Proposition 4. *Let (A_η) be a family of computational algebras. A_η is unconditionally $\approx_E$-ad-sound iff A_η is unconditionally $\approx_E$-sound.*

A direct corollary of this proposition is the following.

Corollary 1. *The implementation of XOR for the signature considered in Section 6.1, (A_η), is unconditionally $\approx_{E_\oplus}$-ad-sound.*

8 Adaptively sound theories

We have already seen that the theory of XOR is unconditionally adaptively sound. We now present additional adaptive soundness results for several equational theories: symmetric encryption (which is adaptively sound under IND-CPA) and modular exponentiation (adaptively sound under DDH). We also consider composed theories: symmetric encryption and modular exponentiation as well as symmetric encryption and XOR. For these theories we allow keys to be computed, using respectively modular exponentiation and XOR. Additional details and proofs can be found in [19].

8.1 Symmetric encryption

We consider the case of probabilistic symmetric encryption which recasts the result of [21] in our framework and illustrates well the difference between a purely passive and an adaptive adversary.

Symbolic model. Our symbolic model consists of the set of sorts $\mathcal{S} = \{Data\}$, an infinite number of names for sort $Data$ called keys and the function symbols:

$\mathrm{enc}, \mathrm{dec} : Data \times Data \to Data$	encrypt, decrypt
$\mathrm{pair} : Data \times Data \to Data$	pair constructor
$\pi_l, \pi_r : Data \to Data$	projections
$\mathrm{samekey} : Data \times Data \to Data$	key equalities test
$\mathrm{tenc}, \mathrm{tpair} : Data \to Data$	type testers
$0, 1 : Data$	constants

A name k is used at a key position in a term t if there exists a sub-term $\mathrm{enc}(t', k)$ of t. Else k is used at a plaintext position. We consider the equational theory E_{sym} generated by:

$$\mathrm{dec}(\mathrm{enc}(x, y), y) = x \qquad \pi_l(\mathrm{pair}(x, y)) = x$$
$$\pi_r(\mathrm{pair}(x, y)) = y \qquad \mathrm{samekey}(\mathrm{enc}(x, y), \mathrm{enc}(z, y)) = 1$$
$$\mathrm{tenc}(\mathrm{enc}(x, y)) = 1 \qquad \mathrm{tpair}(\mathrm{pair}(x, y)) = 1$$

As usual $\mathrm{enc}(t, k)$ is also written $\{t\}_k$ and $\mathrm{pair}(t, t')$ is also written $\langle t, t' \rangle$.

Well-formed frames and adversaries. As usual we forbid the formal terms to contain such cycles. Let $\prec$ be a total order among keys. A *frame φ is acyclic for $\prec$* if for any subterm $\{t\}_k$ of φ, if k' occurs in t then $k' \prec k$. Moreover as noted in [21], selective decommitment [16] can be a problem. The classical solution to this problem is to require keys to be sent *before* being used to encrypt a message or they must never appear as a plaintext. A *frame $\varphi = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is well-formed for $\prec$* if

- φ is acyclic for $\prec$;
- the terms t_i only use symbols enc, pair, 0 and 1, and only names are used at key positions;
- if k is used as plaintext in t_i, then k cannot be used at a key position in t_j for $j < i$.

An *adversary is well-formed for $\prec$* if the sequence of queries $(t_0^i, t_1^i)_{1 \leq i \leq n}$ that he makes to his oracle yields two well-formed frames $\{x_1 \mapsto t_0^1, \ldots, x_n \mapsto t_0^n\}$ and $\{x_1 \mapsto t_1^1, \ldots, x_n \mapsto t_1^n\}$ for $\prec$.

Concrete model. The family of computational algebras (A_η) giving the concrete semantics depends on a symmetric encryption scheme $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$. The concrete domain $[\![Data]\!]_{A_\eta}$ contains all the possible bitstrings and is equipped with the distribution induced by $\mathcal{KG}$. Interpretation for constants 0 and 1 are respectively bitstrings 0^η and 1^η. The enc and dec function are respectively interpreted using algorithm $\mathcal{E}$ and $\mathcal{D}$. We assume the existence in the concrete model of a concatenation operation which is used to interpret the pair symbol. The corresponding left and right projections implement π_l and π_r. Finally, as we are only interested in well-formed frames, we do not provide any computational interpretation for tenc, tpair and $\mathrm{samekey}$.

Semantic security. In this section we suppose a message-length, but not necessarily which-key concealing semantically secure encryption scheme. The definition that we recall below uses a left-right encryption oracle $LR^b_{\mathcal{SE}}$. This oracle first generates a key k using $\mathcal{KG}$. Then it answers queries of the form (bs_0, bs_1), where bs_0 and bs_1 are bitstrings. The oracle returns ciphertext $\mathcal{E}(bs_b, k)$. The goal of the adversary $\mathcal{A}$ is to guess the value of bit b. His advantage is defined as:

$$\mathrm{Adv}^{cpa}(\mathcal{A}, \eta, \mathcal{SE}) = \mathbb{P}\left[\mathcal{A}^{LR^1_{\mathcal{SE}}} = 1\right] - \mathbb{P}\left[\mathcal{A}^{LR^0_{\mathcal{SE}}} = 1\right]$$

Encryption scheme $\mathcal{SE}$ is IND-CPA secure if the advantage of any adversary $\mathcal{A}$ is negligible in η. The standard definition of IND-CPA allows the scheme to be message-length revealing. By abuse of notation we call the above scheme also IND-CPA secure.

We also describe a variant of IND-CPA security, IND-CPA′, which models non-adaptive adversaries. The left-right encryption oracle $LR'^b_{\mathcal{SE}}$ takes as input a list of pairs of bitstrings (bs_0^i, bs_1^i) for i in $[1, n]$ and returns the list of ciphertexts $\mathcal{E}(bs_b^i, k)$ for i in $[1, n]$. This oracle can only be queried once. The adversary can observe multiple encryptions but he is not allowed to chose them adaptively. The advantage of an adversary is defined in a similar way as above, replacing $LR^b_{\mathcal{SE}}$ by $LR'^b_{\mathcal{SE}}$. A symmetric encryption scheme is said to be IND-CPA′ if the advantage of any polynomial time adversary $\mathcal{A}$ is negligible in η. These two notions of semantic security are related by the following proposition.

Proposition 5. *Let $\mathcal{SE}$ be a symmetric encryption scheme. If $\mathcal{SE}$ is* IND-CPA*, then $\mathcal{SE}$ is* IND-CPA′*. However $\mathcal{SE}$ can be* IND-CPA′ *without being* IND-CPA.

We now state the soundness theorem for symmetric encryption.

Theorem 5. *Let $\prec$ be a total order among keys. In the remainder of this proposition we only consider well-formed adversaries for $\prec$. Let (A_η) be a family of computational algebras based on a symmetric encryption scheme $\mathcal{SE}$.*

- *(A_η) is $\approx_{E_{\mathsf{sym}}}$-ad-sound if $\mathcal{SE}$ is* IND-CPA *but the converse is false.*
- *(A_η) is $\approx_{E_{\mathsf{sym}}}$-sound if $\mathcal{SE}$ is* IND-CPA′ *but the converse is false.*

The proof uses a similar hybrid argument as the one used by Micciancio and Panjwani in [21]. Results of this section are summed up in the following table. Note that the relations between adaptive and non-adaptive soundness have not been detailed formally.

$$\begin{array}{ccc} \approx_{E_{\mathsf{sym}}}\text{-ad-sound} & \begin{array}{c}\Leftarrow\\ \not\Rightarrow\end{array} & \text{IND-CPA} \\ \not\Uparrow \quad \Downarrow & & \not\Uparrow \quad \Downarrow \\ \approx_{E_{\mathsf{sym}}}\text{-sound} & \begin{array}{c}\Leftarrow\\ \not\Rightarrow\end{array} & \text{IND-CPA}' \end{array}$$

8.2 Modular exponentiation

We suppose the same symbolic and concrete model as in Section 6.2. The DDH assumption is necessary and sufficient to prove adaptive soundness.

Theorem 6. *Let (A_η) be a family of computational algebras. (A_η) is $\approx_{E_{\mathsf{DH}}}$-sound iff (A_η) satisfies the* DDH *assumption. (A_η) is $\approx_{E_{\mathsf{DH}}}$-ad-sound iff (A_η) satisfies the* DDH *assumption.*

The proof of this result uses an adaptive variant of DDH called 3DH [12]: it generalizes several previously used variants of DDH. The main difficulty in this proof consists in relating DDH and 3DH.

Results for modular exponentiation are summed up in the following table. Note that while adaptive soundness and (classical) soundness are not equivalent for symmetric encryption, they coincide in this case.

$$\approx_{E_{\mathsf{DH}}}\text{-ad-sound} \iff \text{DDH} \iff \approx_{E_{\mathsf{DH}}}\text{-sound}$$

8.3 Composing encryption with exponentiation

Symbolic model. We consider an equational theory E containing both E_{DH} and E_{sym} and suppose that G is a subsort of $Data$.

Well-formed frames. Let $\prec$ be a total order between keys and exponentiations. A frame φ (on Σ) is well-formed for $\prec$ if:

- φ does not contain any dec, tenc, tpair, π_l, π_r or $*$ symbol, only names and exponentiations are used at key position.
- For any subterm $\exp(p)$ of φ used at a key position, p is linearly independent of other polynomials p' such that $\exp(p')$ is a subterm of φ.
- For any subterm $\{t\}_{t'}$ of φ, if t'' is a name of sort $Data$ or an exponentiation then $t'' \prec t'$.

The second condition is similar to the conditions on key cycles. The last condition is to avoid selective decommitment.

Concrete model. The concrete model is given by the models for symmetric encryption and modular exponentiation. However, exponentiations can be used as symmetric keys in our symbolic model. This needs to be reflected in the concrete model. The family of computational algebras (A_η) giving the concrete semantics is parameterized by a symmetric encryption scheme $\mathcal{SE}$ and an instance generator IG. We require the key generation algorithm of $\mathcal{SE}$ to randomly sample an element of $IG(\eta)$. Giving an IND-CPA encryption scheme $\mathcal{SE}'$, we build another IND-CPA encryption scheme $\mathcal{SE}$ which indeed uses such a key generation algorithm. This is achieved by using a *key extractor* algorithm Kex [13]. This algorithm (usually a universal hash function used with the entropy smoothing theorem) is used to transform group elements into valid keys for $\mathcal{SE}'$. Its main characteristic is that applying Kex to a randomly sampled element of a group created by IG produces the same distribution as the one given by the key generation algorithm of $\mathcal{SE}'$. Then the new encryption and decryption algorithms of $\mathcal{SE}$ apply the Kex algorithm to the group element which is used as key. This produces a symmetric key which can be used with the encryption and decryption algorithms of $\mathcal{SE}'$.

The family of computational algebras (A_η) implementing encryption with exponentiation is *EE-secure* if the encryption scheme $\mathcal{SE}$ is secure against IND-CPA and uses a key generation algorithm as described above and IG satisfies the DDH assumption.

Theorem 7. *Let $\prec$ be a total order between keys and exponentiations. Let (A_η) be an EE-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

8.4 Composing encryption with XOR

Symbolic model. We consider an equational theory E containing both $E_\oplus$ and E_{sym} and suppose that $Data_\oplus$ is a subsort of $Data$.

Well-formed frames. Let $\prec$ be a total order between keys and terms of sort $Data_\oplus$. A frame $\varphi = \{x_1 \mapsto t_1, \dots x_n \mapsto t_n\}$ is well-formed for $\prec$ if the following conditions are verified. Let X be the set of maximal subterms of φ of sort $Data_\oplus$[2].

[2] Using standard definitions for manipulating terms X is formally defined as follows: $X = \bigcup_{1 \le i \le n} \{t_i|_p \mid p \in pos(t_i), sort(t_i|_p) = Data_\oplus, p = p' \cdot k \Rightarrow sort(t_i|_{p'}) \neq Data_\oplus\}$.

– φ does not contain function symbols dec, tenc, tpair, π_l or π_r and only terms of sort $Data_\oplus$ and names are used at key positions.
– For any $x \in X$ used at a key position, there does not exist a set $\{x_1, \ldots, x_i\} \subseteq X \cup \{1\}$, disjoint from $\{x\}$, such that $x =_{E_\oplus} x_1 \oplus ... \oplus x_i$.
– For any subterm $\{t\}_{t'}$ of φ, if t'' is a subterm of t which is a name of sort $Data$ or an element of X then $t'' \prec t'$.

Concrete model. The concrete model is given by the models for symmetric encryption and exclusive OR. However, as in the combination of encryption with exponentiation, we need to reflect that nonces can be used as keys. The family of computational algebras (A_η) giving the concrete semantics is parameterized by a symmetric encryption scheme $\mathcal{SE}$. The XOR part uses the same implementation as in Section 6.1. We require that the key generation algorithm of $\mathcal{SE}$ consists in randomly sampling an element of $[0,1]^\eta$. The family of computational algebras (A_η) is said *EX-secure* if the encryption scheme $\mathcal{SE}$ is secure against IND-CPA and uses a key generation algorithm as described above.

Theorem 8. *Let $\prec$ be a total order between keys and terms of sort $Data_\oplus$. Let (A_η) be an EX-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

9 Conclusion

In this paper we have described computationally soundness results for a model relying on equational theories and static equivalence. We consider the case of passive and adaptive adversaries and present several examples of sound equational theories to illustrate this framework. Whether this framework can be generalized to an active attacker is still a challenging research topic.

References

1. M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In *Proc. 9th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'06)*, volume 3921 of *LNCS*, 2006.
2. M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *LNCS*, pages 46–58, 2004.
3. M. Abadi and C. Fournet. Mobile values, new names, and secure communications. In *Proc. 28th Annual ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.
4. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS'00)*, volume 1872 of *LNCS*, pages 3–22, 2000.
5. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.
6. P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS'05)*, volume 3679 of *LNCS*, pages 374–396, 2005.

7. P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *LNCS*, pages 374–396. Springer, 2005.
8. P. Adão, G. Bana, and A. Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *Proc. 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 170–184, 2005.
9. M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *Proc. 10th ACM Conference on Computer and Communications Security (CCS'03)*, 2003.
10. M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 652–663. Springer, 2005.
11. B. Blanchet. Automatic proof of strong secrecy for security protocols. In *Proc. 25th IEEE Symposium on Security and Privacy (SSP'04)*, pages 86–100, 2004.
12. E. Bresson, Y. Lakhnech, L. Mazaré, and B. Warinschi. A generalization of ddh with applications to protocol analysis and computational soundness. In *Advances in Cryptology - CRYPTO'07, Proc. 27th Annual International Cryptology Conference*, volume 4622, pages 482–499. Springer, 2007.
13. O. Chevassut, P.-A. Fouque, P. Gaudry, and D. Pointcheval. Key derivation and randomness extraction. Technical Report 2005/061, Cryptology ePrint Archive, 2005. http://eprint.iacr.org/.
14. R. Corin, J. Doumen, and S. Etalle. Analysing password protocol security against off-line dictionary attacks. *ENTCS*, 121:47–63, 2005.
15. V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.
16. C. Dwork, M. Naor, O. Reingold, and L. J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.
17. F. D. Garcia and P. van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In *Advances in Information and Computer Security (IWSEC'06)*, volume 4266 of *LNCS*, pages 33–47. Springer, 2006.
18. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
19. S. Kremer and L. Mazaré. Adaptive soundness of static equivalence. In *Proc. 12th European Symposium on Research in Computer Security (ESORICS'07)*, volume 4734 of *LNCS*, pages 610–625. Springer, Sept. 2007.
20. P. Laud and R. Corin. Sound computational interpretation of formal encryption with composed keys. In *Proc. 6th International Conference on Information Security and Cryptology (ICISC'03)*, volume 2971 of *LNCS*, pages 55–66. Springer, 2004.
21. D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Proc. 2nd Theory of cryptography conference (TCC'05)*, volume 3378 of *LNCS*, pages 169–187. Springer, 2005.
22. D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *Journal of Computer Security*, 12(1):99–129, 2004.
23. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proc. 1st Theory of Cryptography Conference (TCC'04)*, volume 2951 of *LNCS*, pages 133–151. Springer, 2004.
24. D. H. Phan and D. Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In *Proc. Selected Areas in Cryptography (SAC'04)*, volume 3357 of *LNCS*, pages 185–200, 2004.