Computational soundness of static equivalence

Véronique Cortier[1], Steve Kremer[2], and Pascal Lafourcade[3]

[1] LORIA, CNRS & INRIA
[2] LSV, CNRS & ENS Cachan & INRIA
[3] Verimag, CNRS & Université Grenoble 1

Abstract. Privacy related properties in electronic voting are naturally expressed as indistinguishability properties. This motivates the study of observational equivalence, as well as static equivalence in the context of the AVOTÉ project. In this report we survey the existing results on the *computational soundness* of symbolic indistinguishability relations in the presence of a passive adversary, for which several results were obtained by the members of the AVOTÉ project. This report is based on a recent survey [CKW09] on computational soundness of symbolic methods for analysing security protocols, carried out in the context of the AVOTÉ project.

1 Introduction

Security protocols are short distributed programs designed to achieve various security goals, such as data privacy and data authenticity, even when the communication between parties takes place over channels controlled by an attacker. Their ubiquitous presence in many important applications makes designing and establishing the security of cryptographic protocols a very important research goal. Two distinct approaches that have evolved starting with the early 1980's attempt to ground security analysis of protocols on firm, rigorous mathematical foundations. These two approaches are known as the computational (or the cryptographic) approach and the symbolic (or the Dolev-Yao, or the formal methods) approach. Each approach relies on mathematical models for the executions of protocols/primitives in adversarial environments, formally define security properties expected from cryptographic systems, and develop methods for rigorously proving that given constructions meet these requirements.

The central features of the computational approach are detailed, bit-level models for system executions and a powerful adversary: security is assessed against *arbitrary* probabilistic polynomial time machines. It is generally acknowledged that security proofs in this model offer powerful security guarantees. A serious downside of this approach however is that proofs for even moderately-sized protocols are usually long, difficult, tedious, and highly error prone.

In contrast, symbolic methods employ a highly abstract view of the execution where the messages exchanged by parties are symbolic terms. Furthermore, primitives are assumed absolutely secure, which in turn leads to severe restrictions on the power of the adversary. For instance, it is postulated the plaintext underlying a ciphertext can only be recovered if the adversary has or can derive the appropriate decryption key. The resulting models are considerably simpler than those of the computational approach, proofs are therefore also simpler, and can sometimes benefit from machine support. An important problem with this approach is that the high level of abstraction renders unclear the security guarantees that this approach offers.

Due perhaps to the widely different set of tools and techniques, the two approaches have coexisted and developed independently for many years. The lack of interaction between the two communities also meant that the relation between models, security results and guarantees using the two approaches was only superficially understood. Abadi and Rogaway were the first to demonstrate that establishing close relations between the models is not only possible, but also that it holds significant promise. Through their work it became clear that it is possible to employ the tools and methods specific to the symbolic approach to directly obtain computational security guarantees. The crucial implication is that such guarantees can be obtained without making use of the typical computational proofs. This realization motivated a significant amount of follow-up work. In this report we first recall the seminal result by Abadi and Rogaway and then discuss several extensions to this result. In particular we will discuss results on static equivalence which generalizes the pattern equivalence used by Abadi and Rogaway. The results are also summarized in Table 1.

Several of these results have been obtained in the context of the AVOTÉ project:

- Extension of the Abadi-Rogaway approach for bilinear pairings [KM09];
- A framework for proving cryptographic indistiguishability through symbolic static equivalence, with application to lists and cyphers [BCK09];
- A general framework for relating formal and computational models for generic encryption schemes in the random oracle model [ELN09].

For each of these results, the corresponding publication is appended to this report.

2 The Abadi-Rogaway result

The result of Abadi and Rogaway shows that if a symbolic notion of secrecy of data that occurs in a message is satisfied, then a computational notion is also satisfied [AR00,AR02]. Their result holds for a class of messages constructed as in the following section.

Formal expressions and equivalence. On the formal side, one considers a simple grammar for expressions. 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.

$$\begin{array}{ll} M, N ::= & \quad \textit{expressions} \\ \quad K & \text{key } (K \in \mathbf{Keys}) \\ \quad i & \text{bit } (i \in \mathbf{Bool}) \\ \quad \langle M, N \rangle & \text{pair} \\ \quad \{M\}_K & \text{encryption } (K \in \mathbf{Keys}) \end{array}$$

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 $\Box$. This idea is formally captured by the following function p. The function takes as arguments a term and a set T of keys and is defined inductively as follows.

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

The pattern of an expression M is defined by

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

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

Furthermore, expressions M and N are formally indistinguishable, written $M \equiv N$ if and only if $pattern(M) = pattern(N)\sigma$, where σ is a bijective renaming of keys. For example, we have that $0 \not\equiv 1$, $K_0 \equiv K_1$, $\{0\}_{K_1} \equiv \{1\}_{K_0}$ and $\langle K_0, K_0\rangle \not\equiv \langle K_0, K_1\rangle$.

Computational setting and hypotheses on the implementation. In the computational setting, one reasons at the level of bitstrings and algorithms executed on Turing Machines, rather than on abstract terms. Expressions are interpreted as bitstrings by instantiating each of the symbolic operations (including the constants) via appropriate algorithms. In particular we assume a computational pairing function that takes as input two bitstrings m_1 and m_2 and outputs their concatenation $\langle m_1, m_2\rangle$. The function is such that m_1 and m_2 are easily extractable from $\langle m_1, m_2\rangle$. Furthermore, we use a concrete encryption scheme, which is a triple of polynomial time algorithms $\mathcal{K}, \mathcal{E}, \mathcal{D}$ for key generation, encryption and decryption respectively. The key generation algorithm is parameterized by a security, or complexity parameter $\eta \in 1^*$. Intuitively, η defines the key length. As expected we require that $\mathcal{D}_k(\mathcal{E}_k(m, r)) = m$ for any $k \in \mathcal{K}(\eta)$, message m, and random bitstring r (that represents the coins of the probabilistic encryption algorithm).

The Abadi-Rogaway result relies on a security notion for encryption schemes termed "type-0" in the original paper [AR00]. Here, we call schemes that satisfy this notion, which we recall bellow, simply *secure*. Informally, secure schemes hide all information about encrypted plaintexts (including their length) and hide all information about the encryption key. This notion is significantly stronger than more standard ones which allow for ciphertexts to reveal the length of the underlying plaintext as well as partial information about the encryption key. The stronger assumption is used for simplicity as the Abadi-Rogaway framework can be further refined to only rely on the more standard notions.

An encryption scheme is *secure* if for any security parameter η and any probabilistic polynomial time Turing machine $\mathcal{A}$ (the adversary) the advantage

$$\begin{aligned} Adv(\mathcal{A}) = \Pr[k, k' \xleftarrow{R} \mathcal{K}(\eta) : \mathcal{A}^{\mathcal{E}_k(\cdot), \mathcal{E}_{k'}(\cdot)}(\eta) = 1] - \\ \Pr[k \xleftarrow{R} \mathcal{K}(\eta) : \mathcal{A}^{\mathcal{E}_k(0), \mathcal{E}_k(0)}(\eta) = 1] \end{aligned}$$

is a negligible function of η. Here, $x \xleftarrow{R} \mathcal{D}$ denotes the random sampling of an element of distribution $\mathcal{D}$ and $\mathcal{A}^{\mathcal{O}}$ is the Turing Machine $\mathcal{A}$ that has access to a set of oracles $\mathcal{O}$. Intuitively, one requires 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 this security under this notion implies that encryption needs to be randomized, so that an adversary does not see identical answers when confronted with the second pair of (identical) oracles. In [AR02], the authors provide constructions for such schemes from standard cryptographic assumption.

A recurrent theme in computational soundness is that of acyclic expressions. The reason is that an encryption scheme respecting the above security definition may be insecure as soon as the adversary is given a *key cycle*. We say that a key K_1 *encrypts* a key K_2 in a formal expression M if M contains a subexpression $\{N\}_{K_1}$ and K_2 occurs in N. In this way any expression M defines a binary relation *encrypts* on keys. We say that an expression contains a key cycle if and only if the corresponding encrypts relation is cyclic. For instance $M_1 = \{K\}_K$ contains a key cycle as K encrypts K. In $M_2 = \{\{K_1\}_{K_2}\}_{K_3}$ we have that K_3 encrypts K_1, K_3 encrypts K_2 and K_2 encrypts K_1 and hence M_2 does not contain any key cycle. In Abadi and Rogaway's main result, key cycles are therefore forbidden. Similar conditions can be found in most soundness results. To better understand the problem of key cycles suppose that $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is a secure encryption scheme and let $\mathcal{SE}' = (\mathcal{KG}', \mathcal{E}', \mathcal{D}')$ be defined as follows:

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

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 secure encryption scheme as it behaves as $\mathcal{SE}$ in nearly all cases (in the security experiment the adversary can 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. $\mathcal{D}$ and $\mathcal{D}'$ are computationally indistinguishable, written $\mathcal{D} \approx \mathcal{D}'$ if for any η and any probabilistic polynomial time Turing machine $\mathcal{A}$, the advantage

$$Adv(\mathcal{A}) = \Pr[x \overset{R}{\longleftarrow} \mathcal{D}_\eta : \mathcal{A}(\eta, x) = 1] - \Pr[x \overset{R}{\longleftarrow} \mathcal{D}'_\eta : \mathcal{A}(\eta, x) = 1]$$

is a negligible function of η.

Interpretation of formal expressions and soundness result. The Abadi-Rogaway result links the notion of pattern equivalence on expressions defined in the previous section with an appropriate notion of computational equivalence defined on distributions. These distributions are associated to expressions using the following algorithms that convert formal expressions into bitstrings.

Bitstrings are tagged using types “key”, “bool”, “pair” and “ciphertext”. The **Initialize** procedure uses $\mathcal{K}$ to generate actual keys for each of the key symbols that occurs in M (that is for each key $K \in Keys(M)$). Then, then **Convert** procedure implements encryption using algorithm $\mathcal{E}$.

```
Initialize_η(M)
    for $K \in Keys(M)$ do $\tau(K) \overset{R}{\longleftarrow} \mathcal{K}(\eta)$

Convert(M)
    if $M = K$ ($K \in \mathbf{Keys}$) then
        return $\langle \tau(K), \text{“key”} \rangle$
    if $M = b$ ($b \in \mathbf{Bool}$) then
        return $\langle b, \text{“bool”} \rangle$
    if $M = \langle M_1, M_2 \rangle$ then
        return $\langle \langle \mathbf{Convert}(M_1), \mathbf{Convert}(M_2) \rangle, \text{“pair”} \rangle$
    if $M = \{M_1\}_K$ then
        $x \overset{R}{\longleftarrow} \mathbf{Convert}(M_1)$
        $y \overset{R}{\longleftarrow} \mathcal{E}_{\tau(K)}(x)$
        return $\langle y, \text{“ciphertext”} \rangle$
```

The **Initialize** and **Convert** procedures associate to a formal term M a family of probability distributions $[\![M]\!] = \{[\![M]\!]_\eta\}$.

Abadi and Rogaway's main result is that for any formal expressions M and N that do not contain key cycles, whenever the computational interpretation of the terms uses a secure encryption scheme (as defined above), then $M \equiv N$ implies that $[\![M]\!] \approx [\![N]\!]$. In other words, they show that pattern-based equivalence is a sound abstraction of cryptographic indistinguishability.

3 Extensions of the Abadi-Rogaway result

The initial result of Abadi and Rogaway has given rise to many extensions. Some of these extensions consider the question of completeness of their logic. Other extensions consider different implementations of encryption (with variants of the initial patterns) as well as other cryptographic primitives.

Completeness of the Abadi-Rogaway logic. In [MW02,MW04], Micciancio and Warinschi show that the Abadi-Rogaway logic is not *complete* as presented in the original paper. Here, by completeness we mean that $M \not\equiv N$ implies that $[\![M]\!] \not\approx [\![N]\!]$, i.e., whenever two formal expressions are not equivalent, then the computational interpretation of these two messages should be distinguishable. Micciancio and Warinschi exhibit a counter-example by constructing a secure encryption scheme and two symbolic expressions that are not symbolically equivalent, which yet give rise to indistinguishable probability distribution ensembles.

They show that completeness can be recovered by implementing encryption with a scheme that is *authenticated.* Informally, an encryption scheme is authenticated if an adversary cannot produce a valid ciphertext different from ciphertexts honestly produced by the parties that posses the encryption key. Gligor and Horvitz [HG03] further refine this completeness result. They introduce a new security criterion for encryption schemes, *weak key-authenticity test for expressions* (WKA-EXP), which is strictly weaker than authenticated encryption. WKA-EXP is both sufficient and necessary for completeness.

Public-key encryption. In [Her03,Her05], Herzog shows a similar result as Abadi and Rogaway, but for public-key encryption. Patterns are generalized in the expected way for expressions that use public-key encryption. The problem of key-cycles also persists in this setting. To define a key-cycle of an expression M in the public-key setting one constructs a graph G_M: the set of vertices is the set of public/private key pairs $\{(pubK_1, privK_1), \dots, (pubK_n, privK_n)\}$; there exists an edge from $(pubK_i, privK_i)$ to $(pubK_j, privK_j)$ if $pubK_i$ encrypts $privK_j$ in M. M has no key-cycle if G_M is acyclic. Herzog presents a soundness theorem, similar to the one of Abadi and Rogaway, whenever the encryption scheme used for the computational interpretation provides indistinguishability under chosen-ciphertext attacks (IND-CCA2 security).

Composed keys. In [LC04], Laud and Corin extend the original soundness theorem to allow arbitrary expressions as keys. The tricky part is again to handle key-cycles correctly. As arbitrary expressions are used in the position of keys, the definition of what is a key cycle is not obvious. Rather than giving an explicit definition of what is a key-cycle, the symbolic adversary is strengthened and the formal equivalence relation directly captures key-cycles. More precisely, an expression is not formally equivalent to its pattern whenever a "key-cycle" is present. For instance, $\{\langle K_1, K_2\rangle\}_{\langle K_1, K_2\rangle} \not\equiv \Box$ and $\langle \{K_1\}_{\langle K_1, K_2\rangle}, \{K_2\}_{\langle K_1, K_2\rangle}\rangle \not\equiv \langle \Box, \Box\rangle$ while $\{K_1\}_{\langle K_1, K_2\rangle} \equiv \Box$, because the second part of the key K_2 does not occur anywhere else.

Handling key cycles. Key cycles have gained a lot of attention in the context of computational soundness. The reason is that there is an inherent difference between their treatment in symbolic models (where such cycles do not cause any troubles) and the computational model (where standard security definitions do not guarantee security in presence of key-cycles.) There are two natural approaches to reconcile this apparent difference.

One possibility is to strengthen the symbolic attacker. This is the direction explored by Laud in [Lau02]. The idea is to modify the symbolic deduction relation so that whenever a key occurs in a key cycle then it becomes known to the attacker. Laud shows an unconditional soundness theorem in the style of Abadi and Rogaway (unconditional in the sense that formal expressions may contain key cycles).

The second possibility is to strengthen the computational notion as to guarantee security even in the presence of key-cycles. This is the approach adopted in [ABHS05], Adão et al. They consider a stronger security notion, called *key-dependent message* (KDM) security which demands security even in the presence of such cycles. They show that soundness holds in a public-key setting in the presence of key cycles when a KDM secure encryption scheme is used for the computational interpretation of encryption. They also prove a separation between standard security notions (IND-CCA2) and KDM security and demonstrate that IND-CCA2 security is not sufficient to provide soundness in the presence of key-cycles. Schemes secure under the KDM notion can be easily constructed in the random oracle model, but schemes secure in the standard model seem much harder to construct. Recently, Boneh et al. [BHHO08] demonstrated the existence of an asymmetric encryption scheme secure under key dependent message attacks in a restricted sense: their scheme does not permit the encryption of messages that depend in arbitrary ways on the set of secret keys.

In most of the other approaches, one has to assume that key cycles cannot be generated, even when the adversary interacts arbitrarily with the protocol. Whether a key cycle can be generated is undecidable in the general case but it has been shown to be NP-complete in the symbolic setting, for an active adversary and a bounded number of sessions [CZ06].

Partial information leakage and information theoretic security. Adão et al. [ABS05] consider different computational implementations of the encryption function. In particular they show soundness and completeness when *which-key* and *length-key revealing* encryption schemes are used. A which-key revealing encryption scheme allows the adversary to detect when two ciphertexts have been encrypted with the same key. At the symbolic level this is reflected by indexing the boxes with the encryption key, yielding a more precise equivalence relation. For instance, $\mathit{pattern}(\{0\}_K) = \Box_K$ and hence we have that $\langle \{0\}_K, \{1\}_K \rangle \not\cong \langle \{0\}_K, \{1\}_{K'} \rangle$. A length-key revealing encryption scheme allows the attacker to learn the length of the plaintext. At the symbolic level the boxes are indexed with the length of the plaintext to reflect this partial information leakage.

The authors also consider the case where encryption is implemented by a one-time pad. Whenever encryption keys are only used once they show that one obtains soundness and completeness with respect to an information-theoretic setting. In such a setting the equivalence is the equality of the probability distributions rather than indistinguishable by a polynomial-time bounded adversary.

Hash functions. Garcia and van Rossum [GvR06] extend the Abadi-Rogaway logic to hash functions. Soundness theorems for hash functions are particularly tricky as in the symbolic model, hash functions do not leak any partial information about the hashed message. Typical computational security definitions for hash functions provide weaker guarantees, such as one-wayness. Garcia and van Rossum show a soundness result when hash functions are implemented using oracle hashing. Oracle hashing has been introduced by Canetti: it is a probabilistic hash function which requires a verification algorithm to check whether a hash corresponds to a given message. These are hash functions that do hide all partial information about the message that is being hashed. In the journal version [GvR08], they extend Micciancio and Warinschi's completeness result to hash functions in a similar way.

Modular exponentiation. Bresson et. al [BLMW07] give an extension of the Abadi-Rogaway logic with modular exponentiation. They show how to extend the notion of patterns in order to capture the information that is leaked through exponentiation, which are essentially linear dependencies between the various exponents. For example, the symbolic secrecy notion captures the idea that an adversary can observe that in the expression (g^x, g^y, g^{2x+y}) the third term can be obtained by squaring the first one and multiplying it with the second. Non-linear relations, as in the expression (g^x, g^y, g^{x+xy}), cannot be observed by the adversary. The soundness for the resulting language relies on a generalization of the Diffie-Hellman assumption which in most relevant cases is implied by the latter.

In the same vein than [BLMW07], Mazaré [Maz07,KM09] presents an extension the Abadi-Rogaway logic with a bilinear pairing operation. Their soundness result assumes the hardness of the bilinear decisional Diffie-Hellman problem and an IND-CPA encryption scheme. The soundness result is illustrated on the Joux tripartite Diffie-Hellman protocol, as well as the TAK-2 and TAK-3 protocols.

Offline guessing attacks. In security protocols passwords or other weak data are often used as encryption keys. For such protocol an important security property is resistance to offline guessing attacks. In such attacks an attacker first collects (possibly by interacting with the protocol) some data. In a second phase, he *guesses* a password out of a dictionary. If the attacker has a means to verify that his guess was correct using the data he had gathered, then the protocol is subject to a guessing, or dictionary attack. In [AW05a], Abadi and Warinschi have shown soundness results for protocols that use password encryptions. They define the computational security of a password encryption primitive: for any two passwords, any polynomially bounded adversary, that is given these two passwords and given access to an oracle, encrypting samples drawn from a plaintext distribution, is not able to distinguish whether the oracle uses the first or the second password for encryption. They also define formal and computational security of expressions against offline guessing attacks in terms of indistinguishability. Then for symmetric, asymmetric and password encryptions with secure implementation they show two soundness theorems. The first one is an extension of the Abadi-Rogaway soundness theorem for indistinguishability. The second theorem states that whenever a formal expression E hides passwords, then its computational interpretation also hides passwords. These results hold for $\mathsf{IND\text{-}CPA}$ secure symmetric and asymmetric schemes, and for password-based encryption schemes that "securely" encrypt keys and ciphertexts of the symmetric and asymmetric schemes. In addition, it only holds for expressions that do not contain key cycles.

Cryptographically controlled access control to XML. A compelling application of computational soundness against passive adversaries was given by Abadi and Warinschi [AW05b,AW08]. The focus of that work is the security of a scheme that uses encryption to enforce access control policies to XML documents. The scheme, designed by Miklau and Suciu [MS03] explains how to obtain from a given XML document and a given access policy a so-called protection: a partially encrypted XML document which enforces the original access policy. The guarantees for the scheme were rather informal.

Abadi and Warinschi formalize the scheme using a symbolic language for expressions that extends the one of Abadi and Rogaway with secret sharing schemes. Then, they show that secrecy as demanded by the policy used to create a certain protection on an XML document is satisfied in a symbolic sense: data that should be secret according to the policy is *symbolically secret* in the expression that describes the protection. It then follows using the computational soundness of the language for expressions that the same data is also computationally secret. The soundness results hold for implementations that use $\mathsf{IND\text{-}CPA}$ encryption schemes and n-out-of-n secure secret sharing schemes.

Soundness against an adaptive adversary. Micciancio and Panjwani [MP05] show a soundness result for encryption and pairing in the presence of a slightly stronger, *adaptive* adversary. Soundness is defined through the following experiment. An adversary has access to a left-right oracle, which given on input two terms M_1 and M_2, returns a sample of the computational interpretation of M_b, where b is the challenge bit of the oracle. The adversary can interact with the oracle but is only allowed to submit queries such that the sequence of queries $(M_1^1, M_2^1), \ldots, (M_1^\ell, M_2^\ell)$ sent to the oracle is such that $\langle M_1^1, \ldots, M_1^\ell \rangle$ is formally equivalent, i.e. has the same pattern up to renaming, to $\langle M_2^1, \ldots, M_2^\ell \rangle$. The adversary wins if he succeeds in outputting b with non-negligible probability. Note that the oracle is stateful and implements terms in a consistent way, i.e. if a key has been drawn in a previous query the same value is reused in subsequent queries. An adaptive adversary is strictly stronger than a purely passive one as he can choose his queries after having already obtained the implementation of some terms. On the technical level, the fact of having an adaptive adversary raises the problem of selective decommitment which is overcome by imposing the following condition: if a key is used to encrypt a message it either must have been sent previously in plaintext or it never appears in plaintext. The usefulness of an adaptive adversary is illustrated by deriving computationally sound symbolic model for the analysis of multicast key distribution protocols. In this model, the adversary cannot directly interact with the protocol participants, but he can influence the control flow.

4 Soundness of static equivalence

Baudet, Cortier, and Kremer have considered a more general alternative to the approach described in the previous sections. They develop a framework in which symbolic secrecy is expressed in terms of *static equivalence*, a well-established equivalence relation from cryptographic pi-calculi[BCK05,BCK09]. This approach is more general in that it does not depend on a particular set of primitives.

Abstract and computational algebras. Independence from a particular primitives is reflected in their use of an arbitrary *abstract algebra* to describe the messages exchanged in a protocol. The algebra is defined over a many-sorted first-order signature equipped with an *equational theory.* For instance, symmetric, deterministic encryption is modeled by the theory E_{enc} generated by the classical equation $dec(enc(x, y), y) = x$. Equality between two terms is generally interpreted modulo the equational theory (denoted $=_E$ for an equational theory E). For example, $dec(enc(m, k), k) =_{E_{enc}} m$. Given an abstract signature a computational algebra A is defined by associating to every sort s of the abstract algebra a set of bitstrings $[\![s]\!]_A \subseteq \{0, 1\}^*$ with an efficient procedure for drawing random elements, and to every function f a computational function $[\![f]\!]_A$. Given a symbolic term T, a distribution $[\![T]\!]_A$ is associated by drawing a random element of the corresponding sort for each name and replacing each function symbol by its computational counterpart.

Security notions, soundness, and faithfulness. The two security notions which are considered are deducibility and static equivalence. Deducibility formalizes which are the terms that an attacker can compute from a given sequence of terms. Static equivalence models whether two sequences of terms can be distinguished. Both deducibility and static equivalence are parameterized by an equational theory. In this approach, static equivalence replaces the pattern-based formal equivalence.

To reason about the soundness of implementations Baudet et al. define soundness for the three relations $=_E$, $\vdash_E$ and $\approx_E$. Soundness of $=_E$ means that whenever two terms are symbolically equal (modulo E), any sample drawn from the distribution implementing those terms should be equal with overwhelming probability. Soundness of $=_E$ is generally a hypothesis which reflects that the equational theory is a reasonable abstraction of the primitives. Similarly, they define soundness for deducibility and static equivalence. When a term is not deducible from a sequence of terms, then an attacker given the distribution implementing the given sequence of terms, should be able to output a sample of the distribution implementing the term with only negligible property. When two sequences of terms are statically equivalent, then the distributions associated to these sequences should be indistinguishable.

Faithfulness of those three relations on the other hand represents a strong version of completeness. Whenever two terms are not equal, a term is deducible or two sequences of terms are not statically equivalent, a computational adversary can show this with overwhelming probability (rather than non-negligible probability which would be completeness). Intuitively, when the relations are faithful, for any symbolic attack there exists an efficient computational attack.

It is shown that for many theories $\approx_E$-soundness implies all other notions of soundness and faithfulness. This emphasizes the importance of $\approx_E$-soundness.

Examples: groups, XOR, ciphers and lists In [BCK05,BCK09], Baudet et al. consider several equational theories to illustrate their framework. First they show the $\approx_E$-soundness of an equational theory modeling groups implies the hardness of several classical cryptographic problems: the discrete logarithm, computational Diffie-Hellman, decisional Diffie-Hellman and RSA problems. Note that this is not a soundness result. It shows that any candidate implementation for $\approx_E$-soundness requires at least the hardness of the usual cryptographic problems. Second, they show the unconditional $\approx_E$-soundness of a theory of XOR. The soundness proof reflects the unconditional security (in the information-theoretic sense) of the One-Time Pad. Finally, they show $\approx_E$-soundness of a theory of ciphers and lists (ciphers are deterministic, length-preserving, symmetric encryption schemes).

Soundness of offline guessing attacks and static equivalence. In [ABW06], Abadi, Baudet and Warinschi use the framework of [BCK05,BCK09] to show $\approx_E$-soundness for an equational theory useful in the context of offline guessing attacks. This theory includes symmetric, and asymmetric encryption as well as pairing. A consequence of this soundness result is its applicability to defining and reasoning about off-line guessing attacks in terms of static equivalence. The result is an intuitively appealing implication to computational security against off-line attacks.

Static equivalence vs formal indistinguishability relations. In [BMS06], Bana, Mohassel and Stegers argue that the notion of static equivalence is too coarse and not sound for many interesting equational theories. They introduce a general notion of formal indistinguishability relation. This highlights that soundness of static equivalence only holds for a restricted set of well-formed frames (in the same vein Abadi and Rogaway used restrictions to forbid key cycles). They illustrate the unsoundness of static equivalence for modular exponentiation.

Formal indistinguishability extended to the ROM In [ELN09] the authors extend Bana et al.'s approach [BMS06], by introducing a notion of symbolic equivalence that allows them to prove security of encryption schemes symbolically. The aim of this work is to prove the security for generic encryption schemes that transform one-way functions to IND-CPA secure encryption schemes. They proposed general definitions of formal indistinguishability relation and formal non-derivability relation, that is symbolic relations that are computationally sound by construction. They extended previous work with respect to several aspects. First, their framework can cope with adaptive adversaries. This is mandatory in order to prove IND-CPA security. Second, many general constructions use one-way functions, and often they are analyzed in the random oracle model: hence the necessity to capture the weak secrecy in the computational world. Third, their closure rules is designed with the objective of minimizing the initial relations which depend of the cryptographic primitives and assumptions. Finally they illustrated their approach on several generic encryption schemes: Bellare and Rogaway in [BR93], Hash El Gamal [BLK00] and the scheme proposed by Pointcheval in [Poi00].

Adaptive soundness of static equivalence. The analogue of [MP05], but for the setting where pattern based equivalence is replaced with static equivalence, has been provided by Kremer and Mazaré [KM07] who extend the framework of [BCK05]. In this case, adaptive soundness is defined through an experiment. The adversary interacts with a left-right oracle, which given two symbolic terms, returns either a sample of the concrete implementation of the first or the second term, according to the oracle's challenge bit. As in [MP05], the adversary is restricted to only provide queries such that the left-hand terms and the right-hand terms form two statically equivalent sequences, rather than pattern-equivalent sequences. They show adaptive soundness of static equivalence for an equational theory modeling modular exponentiation (for a class of well-formed frames, hence not contradicting [BMS06] and under similar assumptions as in [BLMW07]), as well as symmetric encryption with composed keys which can be computed using modular exponentiation or exclusive or.

References

[ABHS05] Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre 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.

[ABS05] Pedro Adão, Gergei Bana, and Andre 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.

[ABW06] Martín Abadi, Mathieu Baudet, and Bogdan 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.

[AR00] Martín Abadi and Phillip 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.

[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.

Soundness of pattern equivalence

– Symmetric encryption [AR00]
– Completeness result [MW02,MW04,HG03]
– Public key encryption [Her03,Her05]
– Symmetric encryption with composed keys [LC04]
– Handling key cycles [Lau02,ABHS05]
– Key dependent message [BHHO08]
– Information-theoretic security [ABS05]
– Hash functions [GvR06] and completeness [GvR08]
– Modular exponentiation [BLMW07] and bilinear pairings [Maz07,KM09]
– Offline guessing attacks [AW05a]
– Cryptographically controlled access to XML [AW05b,AW08]
– Adaptive adversary [MP05]

Soundness of static equivalence

– Framework and application to ciphers, lists and, xor [BCK05,BCK09]
– Offline guessing attacks [ABW06]
– Formal indistinguishability relations [BMS06]
– Formal indistinguishability extended to the ROM [ELN09]
– Adaptive adversary [KM07]

Table 1. Summary of the soundness results.

[AW05a] Martín Abadi and Bogdan Warinschi. Password-based encryption analyzed. In *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 664–676. Springer, 2005.

[AW05b] Martín Abadi and Bogdan Warinschi. Security analysis of cryptographically controlled access to xml documents. In *Proc. 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'05)*, pages 108–117. ACM Press, 2005.

[AW08] Martín Abadi and Bogdan Warinschi. Security analysis of cryptographically controlled access to xml documents. *J. ACM*, 55(2):1–29, 2008.

[BCK05] Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. In *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 652–663. Springer, 2005.

[BCK09] Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. *Information and Computation*, 207(4):496–520, April 2009.

[BHHO08] Dan Boneh, Shai Halevi, Mike Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, 2008.

[BLK00] Joonsang Baek, Byoungcheon Lee, and Kwangjo Kim. Secure length-saving elgamal encryption under the computational diffie-hellman assumption. In Ed Dawson, Andrew Clark, and Colin Boyd, editors, *Information Security and Privacy, 5th Australasian Conference, ACISP 2000, Brisbane, Australia, July 10-12, 2000, Proceedings*, volume 1841 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2000.

[BLMW07] Emmanuel Bresson, Yassine Lakhnech, Laurent Mazaré, and Bogdan Warinschi. A generalization of ddh with applications to protocol analysis and computational soundness. In *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 482–499. Springer, 2007.

[BMS06] Gergei Bana, Payman Mohassel, and Till Stegers. The computational soundness of formal indistinguishability and static equivalence. In *Proc. 11th Asian Computing Science Conference (ASIAN'06)*, volume 4435 of *LNCS*, pages 182–196. Springer, 2006.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73, New York, USA, November 1993. ACM.

[CKW09] Véronique Cortier, Steve Kremer, and Bogdan Warinschi. A survey of symbolic methods in computational analysis of cryptographic systems. Research Report RR-6912, INRIA, April 2009.

[CZ06] Véronique Cortier and Eugen Zălinescu. Deciding key cycles for security protocols. In *Proc. of the 13th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'06)*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 317–331, Phnom Penh, Cambodia, November 2006. Springer.

[ELN09] Cristian Ene, Yassine Lakhnech, and Van Chan Ngo. Formal indistinguishability extended to the random oracle model. In Michael Backes and Peng Ning, editors, *Computer Security - ESORICS 2009, 14th European Symposium on Research in Computer Security, Saint-Malo, France, September 21-23, 2009. Proceedings*, volume 5789 of *Lecture Notes in Computer Science*, pages 555–570. Springer, 2009.

[GvR06] Flavio D. Garcia and Peter van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In *Proc. International Workshop on Security 2006 (IWSEC'06)*, LNCS, pages 33–47. Springer, 2006.

[GvR08] Flavio D. Garcia and Peter van Rossum. Sound and complete computational interpretation of symbolic hashes in the standard model. *Theoretical Computer Science*, 394:112–133, 2008.

[Her03] Jonathan Herzog. A computational interpretation of dolev-yao adversaries. In *Proceedings of the 3rd IFIP WG1.7 Workshop on Issues in the Theory of Security (WITS'03)*, 2003.

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

[HG03] Omer Horvitz and Virgil D. Gligor. Weak key authenticity and the computational completeness of formal encryption. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *LNCS*, pages 530–547. Springer, 2003.

[KM07] Steve Kremer and Laurent Mazaré. Adaptive soundness of static equivalence. In *Proceedings of the 12th European Symposium on Research in Computer Security (ESORICS'07)*, volume 4734 of *LNCS*, pages 610–625, Dresden, Germany, September 2007. Springer.

[KM09] Steve Kremer and Laurent Mazaré. Computationally sound analysis of protocols using bilinear pairings. *Journal of Computer Security*, 2009. To appear.

[Lau02] Peeter Laud. Encryption cycles and two views of cryptography. In *Nordic Workshop on Secure IT Systems (NORDSEC'02)*, 2002.

[LC04] Peeter Laud and Ricardo 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.

[Maz07] Laurent Mazaré. Computationally sound analysis of protocols using bilinear pairings. In *Proc. 7th International Workshop on Issues in the Theory of Security (WITS'07)*, pages 6–21, 2007.

[MP05] Daniele Micciancio and Saurabh Panjwani. Adaptive security of symbolic encryption. In *Proc. 2nd Theory of Cryptography Conference (TCC'05)*, volume 3378 of *LNCS*, pages 169–187. Springer, 2005.

[MS03] Gerome Miklau and Dan Suciu. Controlling access to published data using cryptography. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 898–909. VLDB Endowment, 2003.

[MW02] Daniele Micciancio and Bogdan Warinschi. Completeness theorems for the abadi-rogaway logic of encrypted expressions. In *Proceedings of the 2nd IFIP WG1.7 Workshop on Issues in the Theory of Security (WITS'02)*, 2002.

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

[Poi00] David Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In *PKC '00: Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography*, pages 129–146, London, UK, 2000. Springer-Verlag.

Computationally Sound Analysis of Protocols using Bilinear Pairings *

Steve Kremer[1] Laurent Mazaré[2]

[1]LSV, ENS Cachan & CNRS & INRIA kremer@lsv.ens-cachan.fr
[2]LexiFi SAS, laurent.mazare@polytechnique.org

Abstract

In this paper, we introduce a symbolic model to analyse protocols that use a bilinear pairing between two cyclic groups. This model consists in an extension of the Abadi-Rogaway logic and we prove that the logic is still computationally sound: symbolic indistinguishability implies computational indistinguishability provided that the Bilinear Decisional Diffie-Hellman assumption holds and that the encryption scheme is IND-CPA secure. We illustrate our results on classical protocols using bilinear pairing like Joux tripartite Diffie-Hellman protocol or the TAK-2 and TAK-3 protocols. We also investigate the security of a newly designed variant of the Burmester-Desmedt protocol using bilinear pairings. More precisely, we show for each of these protocols that the generated key is indistinguishable from a random element.

Keywords: Security, Formal Methods, Dolev-Yao Model, Computational Soundness, Bilinear Pairing

1 Introduction

Recently bilinear pairings such as Weil pairing or Tate pairing on elliptic and hyperelliptic curves have been used to build several cryptographic protocols. One of the first practical pairing-based protocols has been designed by Joux in [29] where a key exchange protocol based on pairing is proposed. This protocol allows three participants to build a shared secret key in a single round. However this protocol was only designed to be secure in the passive setting and is subject to man-in-the-middle attacks. Several key exchange protocols that extend this original protocol were developed, either to ensure some form of authentication [6] or to extend it to a group setting [10]. Pairings were also used

[*] Work partly supported by the ARA SSIA Formacrypt, the ARA SeSur AVOTÉ and the ARTIST2 Network of Excellence.

as a robust building block for other cryptographic primitives such as identity based encryption schemes or signature schemes [21].

Our contributions. In this paper, we propose an extension of the symbolic model from Dolev and Yao [20] for protocols using bilinear pairing and symmetric encryption. To the best of our knowledge, this is the first time pairings are considered in a Dolev-Yao like model. Moreover we prove that our symbolic model is sound in the computational setting: if there are no attacks in the symbolic setting, then attacks in the computational setting have only a negligible probability of success. This is done by extending the Abadi-Rogaway logic from [3] to symbolic terms using pairings. In particular, we need to adapt formal indistinguishability and keep track of linear relations between polynomials which an adversary might use to distinguish terms. The notion of key cycles needs also to be extended in a non-trivial way, again keeping track of linear relations. We use classical cryptographic assumptions from the standard model to prove soundness: the symmetric encryption scheme has to satisfy indistinguishability against chosen-plaintext attacks (IND-CPA) and the bilinear mapping has to satisfy the bilinear decisional Diffie-Hellman assumption (BDDH). The proof also relies on a technical result of independent interest, which states that BDDH implies an extended version of BDDH similar to recent results on DDH [13]. Under these assumptions, our soundness result can be used to prove computational security of protocols such as Joux tripartite Diffie-Hellman protocol [29] or the TAK-2 and TAK-3 protocols from Al-Riyami and Paterson [6]. By computational security we mean that the generated key is indistinguishable from a random element. To illustrate the scope of our result we also design a new pairing based variant of the Burmester-Desmedt [14] protocol and prove its security in the passive setting.

We stick to the passive setting of [3]. This setting is restrictive compared to results for active adversaries. However this restriction can be partially removed. As shown by Katz and Yung [30], it is possible to automatically transform a (key agreement) protocol that is secure in the passive setting into a protocol that is secure in the active setting. Hence a protocol that is provably secure against active adversaries can be designed using the following methodology: *(i)* design a protocol and prove that it is secure against a passive, symbolic adversary; *(ii)* use the soundness result of this paper to conclude that this protocol is secure against passive adversaries in the computational setting; *(iii)* apply the Katz and Yung compiler to generate a protocol that is secure against active adversaries in the computational setting.

Related work. This result follows the line of a recent trend in bridging the gap which separates the symbolic and computational views of cryptography. This work started with [3, 2] where only passive adversaries are considered.

Further work focused on extending this result by considering the active setting and by adding cryptographic primitives. The active setting has been explored through a rich and generic framework by Backes et al. in [7] and subsequent papers. Micciancio and Warinschi later proposed another soundness result for the active case in [34]. They consider a less general framework but in their model automatic verification of protocols in the symbolic model is possible through existing tools. This model was later extended in [18, 28] in order to remove some of the original limitations and to consider digital signatures. The work of Canetti and Herzog [15] shows that symbolic proofs obtained by the tool ProVerif imply universal composable security for a restricted class of key exchange protocols.

In the passive setting, numerous cryptographic primitives have been studied. Baudet et al. [11] consider exclusive or and ciphers. Low entropy passwords which are subject to guessing attacks are studied in [1]. Garcia and van Rossum [22] prove soundness of symbolic hashes by using probabilistic hash functions. In [4] a stronger variant of semantic security is used to allow symmetric encryption schemes in the presence of key cycles. Adão et al. [5] allow symmetric encryption which leaks partial information about the length and the key. Laud and Corin [31] did consider composed keys. There have also been results on completeness of symbolic models [33, 26, 5, 11]. However we are not aware of any computational soundness result involving pairing-based protocols.

Variants of the classical Diffie-Hellman assumption are used to characterize the security of bilinear pairings [29]. Hence the concept and difficulties of considering pairings are close to those introduced by considering Diffie-Hellman exponentiation. But computational soundness for this primitive has only been considered in a few works. In [24, 19, 35, 36], results for protocols based on Diffie-Hellman exponentiation are given for the computational protocol composition logic. Herzog presents in [25] an abstract model for Diffie-Hellman key exchange protocols; however in this work the abstract model is very different from classical Dolev-Yao models for modular exponentiation [16] as the adversary is extended with the capability of applying arbitrary polynomial time functions. Bana et al. discuss some of the difficulties to obtain computational soundness for Diffie-Hellman exponentiation in [8]. More recently, Bresson et al. [13] extended the computational soundness result of Abadi and Rogaway [3] to Diffie-Hellman exponentiation. This result relies on a powerful generalization of the Decisional Diffie-Hellman (DDH) assumption and its equivalence with the original DDH assumption. However pairings are not considered in their work, neither in the computational soundness result, nor in the generalization of DDH.

Outline of this paper. The next section recalls the necessary definition for bilinear pairings and introduces BDDH security. Section 3 details our symbolic model: terms, deducibility and equivalence are defined in this setting. In section 4 we present our computational setting by giving concrete semantics to symbolic terms. Our main soundness result is given in section 5: symbolic indistinguishability implies computational indistinguishability for secure cryptographic primitives. Section 6 illustrates this soundness result on some simple protocols using bilinear pairings. Finally a short conclusion is drawn in section 7.

2 Preliminaries on Bilinear Pairings

In this section, we briefly recall the basics of bilinear pairings. The formal definition is given in section 4. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of same prime order q. Let g_1 be a generator of $\mathbb{G}_1$. We use multiplicative notations for both groups. A mapping e from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_2$ is called a *cryptographic bilinear map* if it satisfies the three following properties.

- **Bilinearity**: $e(g_1^x, g_1^y) = e(g_1, g_1)^{xy}$ for any x, y in $\mathbb{Z}_q$.
- **Non-degeneracy**: $e(g_1, g_1)$ is a generator of $\mathbb{G}_2$ which is also denoted by g_2, *i.e.*, $g_2 \neq 1_{\mathbb{G}_2}$.
- **Computable**: there exists an efficient algorithm to compute $e(u, v)$ for any u and v in $\mathbb{G}_1$.

Examples of cryptographic bilinear maps include modified Weil pairing [12] and Tate pairing [9]: $\mathbb{G}_1$ is a group of points on an elliptic curve and $\mathbb{G}_2$ is a multiplicative subgroup of a finite field. The traditional notation for group $\mathbb{G}_1$ originates from elliptic curve groups and thus is additive. However we prefer a multiplicative notation in order to simplify our symbolic model of section 3.

The classical decisional security assumption for groups with pairing is the Bilinear Decisional Diffie-Hellman (BDDH) assumption. This assumption states that it is difficult for an adversary that has access to three elements of $\mathbb{G}_1$, g_1^x, g_1^y and g_1^z to distinguish g_2^{xyz} from a randomly sampled element g_2^r of $\mathbb{G}_2$.

A simple key exchange protocol has been proposed by Joux in [29]. This protocol is an extension of the classical Diffie-Hellman key exchange for three participants. Let A, B and C be the three participants. Each of them randomly samples a value in $\mathbb{Z}_q$ (denoted by x for A, by y for B and by z for C). Then the three following messages are exchanged:

$$(1)\ A \rightarrow B, C : g_1^x \qquad (2)\ B \rightarrow A, C : g_1^y \qquad (3)\ C \rightarrow A, B : g_1^z$$

The shared secret key is g_2^{xyz}. It is easy to check that A, B and C can compute this key by using the bilinear map e on the two messages that they have received. Security of this protocol with respect to key indistinguishability in the passive setting is identical to the BDDH assumption [29]. No form of authentication is provided in this protocol, so it is trivially subject to man-in-the-middle attacks.

In the following sections, our objective is to provide a symbolic model for protocols using bilinear maps and to give a computational justification of this model. We stick to the passive setting but as noted earlier this is not a real restriction thanks to the Katz and Yung compiler [30]. As usual the computational setting is parameterized by a security parameter η which can be thought of as the key length. Adversaries are probabilistic polynomial-time (in η) Turing machines. In this paper, we suppose that the adversaries are given implicit access to as many fresh random coins as needed, as well as to the complexity parameter η.

3 The Symbolic Setting

In this section, we introduce the symbolic view of cryptography: messages are represented as algebraic terms, the adversary's capabilities are defined by an entailment relation $\vdash$ and an observational equivalence $\cong$. This equivalence is an extension of the well-known Abadi-Rogaway logic to terms using symmetric encryption and pairing. The main difference with the original logic is that we introduce generator g_1 for the first group ($\mathbb{G}_1$), and generator g_2 for the second group ($\mathbb{G}_2$) as well as an infinite set of names representing exponents.

3.1 Terms and Deducibility

Let **Keys** and **Exponents** be two countable disjoint sets of symbols for keys and exponents. A power-free 3-monomial is a product of three *distinct* exponents and a power-free 3-polynomial is a linear combination of monomials using coefficients in $\mathbb{Z}$ (with no constant coefficient). Hence let x_1, x_2, x_3, x_4 and x_5 be five distinct elements of **Exponents**, $2x_1x_2x_3 + x_3x_4x_5$ is a power-free 3-polynomial but $x_1^2x_2$ and $x_1x_2x_3+1$ are not. We let **Poly** be the set of power-free 3-polynomials with variables in **Exponents** and coefficients in $\mathbb{Z}$. With a slight abuse of notation, we often refer to power-free 3-monomials as monomials and to power-free 3-polynomials as polynomials. Our symbolic setting is restricted to 3-monomials because this is the classical way to use bilinear pairing; using pairings with monomials of order different than 3 might be unsafe.

Let k, x and p be meta-variables over **Keys**, **Exponents** and **Poly** respectively. Polynomials can be used as exponents and the set $\mathbf{T}$ of terms is built using symbolic encryption and concatenation of keys, exponents and exponentiations:

$$\begin{array}{rcl} msg & ::= & (msg, msg) \mid \{msg\}_{key} \mid x \mid key \mid g_1^x \\ key & ::= & k \mid g_2^p \end{array}$$

Term (t_1, t_2) represents the pair composed of terms t_1 and t_2, $\{t\}_k$ represents (symmetric) encryption of term t using key k. In the remainder of the paper we will sometimes use tuples instead of nested pairs in order to simplify the notation. $\{t\}_{g_2^p}$ represents encryption of term t using a key derived from g_2^p (in the computational semantics we assume the implicit application of a deterministic key extraction algorithm Kex which is detailed below). g_1^x and g_2^p represent modular exponentiation of g_1 (generator of the first group) and g_2 (generator of the second group) to the power of an exponent x in the first case and a polynomial p in the second case.

We use classical notations for manipulating terms. A position is a finite word over the natural numbers, ϵ denotes the empty word and $w_1 \cdot w_2$ is the concatenation of w_1 and w_2. The set of positions $pos(t)$ of a term t is inductively defined as $pos(x) = pos(k) = pos(g_i) = \{\epsilon\}$ and $pos(f(t_1, t_2)) = \{\epsilon\} \cup \bigcup_{i \in \{1,2\}} i \cdot pos(t_i)$ where f represents either pairing, encryption or exponentiation. If p is a position of t then the expression $t|_p$ denotes the subterm of t at the position p, *i.e.*, $t|_\epsilon = t$ and $f(t_1, t_2)|_{i \cdot p} = t_i|_p$.

Example 3.1 *Let $t = (k, \{k'\}_k)$. The set of positions pos(t) of t is $\{\epsilon, 1, 2, 2 \cdot 1, 2 \cdot 2\}$. Moreover, $t|_1 = t|_{2\cdot 2} = k$ and $t|_{2\cdot 1} = k'$.*

We say that g_2^p *occurs at a key position* in term t if $\{t'\}_{g_2^p}$ is a subterm of t for some t'. Otherwise we say that g_2^p *occurs as data.* Note that in a same term g_2^p may occur both at a key position and as data. An exponent x can be used as an exponent of either g_1 (*e.g.*, in term g_1^x) or g_2 (*e.g.*, in term $g_2^{xx_2x_3}$). Otherwise, if x is not used used as an exponent of either g_1 or g_2, we say that x is used as data.

For any term t, $\mathsf{pol}\,(t)$ designates the set of polynomials p such that g_2^p is a subterm of t and $\mathsf{mon}\,(t)$ designates the set of monomials used by polynomials in $\mathsf{pol}\,(t)$.

Example 3.2 *Let $t = (\{k\}_{g_2^{2x_1x_2x_3+x_4x_5x_6}}, g_1^{x_1}, g_2^{x_1x_2x_3})$. Then $\mathsf{pol}\,(t) = \{2x_1x_2x_3 + x_4x_5x_6, x_1x_2x_3\}$ and $\mathsf{mon}\,(t) = \{x_1x_2x_3, x_4x_5x_6\}$.*

Equality between polynomials is considered modulo the classical equational theory: associativity and commutativity for addition and multiplication, distributivity of multiplication over addition. Equality can easily be decided, for instance by rewriting polynomials in some normal form $\sum_{i=1}^{n} \lambda_i x_1^{p_{i,1}} \dots x_k^{p_{i,k}}$ and comparing these normal forms.

First we define a deduction relation $E \vdash t$ where E is a finite set of terms and t is a term. The intuitive meaning of $E \vdash t$ is that t can be deduced from E. The deducibility relation is an extension of the classical Dolev-Yao inference system [20]:

$$\frac{t \in E}{E \vdash t} \qquad \frac{E \vdash (t_1, t_2)}{E \vdash t_1} \qquad \frac{E \vdash (t_1, t_2)}{E \vdash t_2} \qquad \frac{E \vdash \{t\}_{key} \quad E \vdash key}{E \vdash t}$$

Note that we did not consider *composition rules* such as if t_1 and t_2 are deducible then (t_1, t_2) is also deducible. Indeed these rules are not necessary as deduction is only used to check whether some key can be deduced from a term. As keys are atomic, it is sufficient to consider the four previous rules. By atomic we mean that keys do not include pairs or encryptions but they may obviously be of the form g_2^p. We add four new deduction rules in order to handle pairing. The three first rules correspond to the three possible ways to obtain an exponentiation g_2^{xyz} using the cryptographic bilinear map:

$$\frac{E \vdash x \quad E \vdash g_1^y \quad E \vdash g_1^z}{E \vdash g_2^{xyz}} \qquad \frac{E \vdash x \quad E \vdash y \quad E \vdash g_1^z}{E \vdash g_2^{xyz}} \qquad \frac{E \vdash x \quad E \vdash y \quad E \vdash z}{E \vdash g_2^{xyz}}$$

Note that these three rules correspond to "real" capacities of the adversary in the computational setting. In the first case, an adversary knowing g_1^y and g_1^z can use the bilinear map to produce g_2^{yz}. As he also knows x he can exponentiate g_2^{yz} to obtain g_2^{xyz}. In the second case, the adversary knows y so he can produce g_1^y and act as in the first case. Finally, the third case is also similar, as the adversary can compute g_1^z and act as in the second case.

The fourth rule handles *linear relations* between polynomials.

$$\frac{E \vdash g_2^p \quad E \vdash g_2^q}{E \vdash g_2^{\lambda p+q}} \lambda \in \mathbb{Z}$$

In the computational world an adversary can multiply two group elements g_2^p and g_2^q in order to get g_2^{p+q}. He can also exponentiate a group element g_2^p and obtain $g_2^{\lambda p}$. Thus it is feasible for the adversary to produce $g_2^{\lambda p+q}$ from g_2^p and g_2^q.

Given this deduction relation we can define the set of *deducible keys of term* t as

$$K(t) = \{k \mid t \vdash k\} \cup \{g_2^p \mid t \vdash g_2^p \wedge g_2^p \text{ is a subterm of } t\}$$

After adding the new deductions, the deducibility relation is still decidable.

Proposition 3.3 *Let t be a term and E be a finite set of terms. Then deducibility of t from E is decidable.*

Proof. In this proof, we use the notion of reachability. First remember that a *key term* is either an element k of **Keys** or an exponentiation g_2^q where q is an element of **Poly**.

A subterm t' of t is *reachable* from t using a set K of *key* terms, iff there exists a position p in t such that $t|_p = t'$ and for any prefix p' of p, *i.e.*, $p = p' \cdot p''$, if $t|_{p'}$ is an encryption $\{u\}_{key}$ then $key \in \mathsf{K}$.

We first show that the set $K(t)$ of deducible keys is computable. Note that $K(t)$ is bounded (for inclusion) by the set of keys and exponentiations of t. The set $K(t)$ can be iteratively computed as follows.

1. Initially, K is empty.

   Iterate the following steps until reaching a fix-point:

2. At each step, any key k and exponentiations g_2^p that is reachable in t using keys and exponentiations from K is added to K.

3. We build the set of reachable monomials rm which contains all the monomials $x_1x_2x_3$ from t such that either

   - x_1, x_2 and x_3,
   - or x_1, x_2 and $g_1^{x_3}$,
   - or x_1, $g_1^{x_2}$ and $g_1^{x_3}$

   are reachable in t using K.

4. At the end of each step, if p is a polynomial from $\mathsf{pol}\,(t)$ which is a linear combination of polynomials from K and monomials from rm, then g_2^p is also added to K.

Now let t be a term and E a finite set of terms t_1 to t_n.

1. If t is an atomic key k, then t is deducible if and only if k appears in $K((t_1, \ldots, t_n))$. Thus deducibility is decidable.

2. Else if t is a key g_2^p, then t is deducible if and only if $E, \{k\}_{g_2^p} \vdash k$ where k is a fresh atomic key (*i.e.*, k does not appear in E). Thus deducibility can be decided as in the previous case.

3. Otherwise t is either an exponent, or a pair, or an encryption, or an exponentiation of g_1. As we do not have any composition rule in the definition of $\vdash$, t is deducible if and only if t appears as a subterm in one of the t_j and is reachable using $K((t_1, \ldots, t_n))$. Hence a decision algorithm can first build $K = K((t_1, \ldots, t_n))$ then check for reachability of t in any of the t_j using K.

□

Alternative definitions are possible for the deduction system. For example, we could consider adding the deduction rule $\frac{E\vdash x}{E\vdash g_1^x}$. Then rules $\frac{E\vdash x \quad E\vdash y \quad E\vdash g_1^z}{E\vdash g_2^{xyz}}$ and $\frac{E\vdash x \quad E\vdash y \quad E\vdash z}{E\vdash g_2^{xyz}}$ would not be necessary anymore and the computational soundness results presented later in this document would still be true. However we stick to our deduction system as it reflects in a simple way how a key g_2^p can be deduced from other terms.

Note that we have only shown decidability of the deduction relation. As, in contrast to a computational adversary, a symbolic adversary is not resource-bounded (in particular it is not polynomial-time bounded) we do not need to detail the complexity for our soundness result. From a verification perspective, efficient algorithms are of course needed which would require a more fine-grained complexity analysis of the above procedure.

3.2 Equivalence

Patterns. Patterns are used to characterize the information that can be extracted from a term. These patterns are close to those introduced in [3, 32] but are extended in order to handle modular exponentiation. We introduce a new symbol □ representing a ciphertext that the adversary cannot decrypt. Moreover we consider that the encryption scheme is not necessarily key-concealing. Hence it may be possible for an adversary to observe whether two ciphertexts have been produced using the same key.

Let t be a term and K be a finite set of keys and elements of the second group g_2^p, then the pattern of t using K, $\mathsf{pat}\,(t, \mathsf{K})$ is inductively defined by:

$$\begin{array}{lcll}
\mathsf{pat}\,((t_1, t_2), \mathsf{K}) & = & \big(\mathsf{pat}\,(t_1, K), \mathsf{pat}\,(t_2, \mathsf{K})\big) & \\
\mathsf{pat}\,(\{t'\}_{key}, \mathsf{K}) & = & \{\mathsf{pat}\,(t', \mathsf{K})\}_{key} & \text{if } key \in \mathsf{K} \\
\mathsf{pat}\,(\{t'\}_{key}, \mathsf{K}) & = & \{\Box\}_{key} & \text{if } key \notin \mathsf{K} \\
\mathsf{pat}\,(a, \mathsf{K}) & = & a & \text{for } a \text{ in } x,\ k,\ g_1^x \text{ and } g_2^p
\end{array}$$

The set K is used to store keys that are known by the adversary.

We say that two terms t_1 and t_2 are *equivalent*, $t_1 \equiv t_2$, if they have the same pattern: $t_1 \equiv t_2$ if and only if $\mathsf{pat}\,(t_1, K(t_1)) = \mathsf{pat}\,(t_2, K(t_2))$. Intuitively patterns hide information that are encrypted with undeducible keys. Hence two terms have the same pattern if the information that can be extracted is the same, so it is impossible to distinguish these two terms.

Equivalence up to renaming. We allow (bijective) renaming of keys in a similar way as [3] but renaming of polynomials is slightly more complex and relies on a linear relation preserving bijection between polynomials. Let us illustrate this on the two following examples.

- Let t_1 be the term $(x_1, x_2, g_1^{x_3}, g_2^{x_4x_5x_6}, g_2^{x_1x_2x_3+x_4x_5x_6})$ and t_2 be the term $(x_1, x_2, g_1^{x_3}, g_2^{x_4x_5x_6}, g_2^{x_7x_8x_9})$. A bijection from polynomials of t_2 to polynomials of t_1 could be

  $$\{x_7x_8x_9 \mapsto x_1x_2x_3 + x_4x_5x_6 \ ; \ x_4x_5x_6 \mapsto x_4x_5x_6\}$$

  However this bijection does not correctly preserve linear relations. In term t_1, $g_2^{x_1x_2x_3+x_4x_5x_6}$ can be obtained by multiplying $g_2^{x_4x_5x_6}$ with $g_2^{x_1x_2x_3}$ (which is obtained by applying the bilinear map to $g_1^{x_2}$ and $g_1^{x_3}$ and raising the result to the power x_1). In term t_2, $g_2^{x_7x_8x_9}$ cannot be obtained in a similar way.

- Let t_1 be the term $(g_2^{x_4x_5x_6}, g_2^{x_1x_2x_3+x_4x_5x_6})$ and t_2 be the term $(g_2^{x_4x_5x_6}, g_2^{x_7x_8x_9})$. The associated bijection is

  $$\{x_7x_8x_9 \mapsto x_1x_2x_3 + x_4x_5x_6 \ ; \ x_4x_5x_6 \mapsto x_4x_5x_6\}$$

  This bijection correctly preserves linear relations as $g_2^{x_1x_2x_3+x_4x_5x_6}$ cannot be obtained from other parts of t_1 ($x_1x_2x_3 + x_4x_5x_6$ is not involved in any linear relations) and $g_2^{x_7x_8x_9}$ cannot be obtained from other parts of t_2.

In order to properly define what is a linear relation preserving bijection, we first introduce the set $dm(t)$ of *deducible monomials* from t, *i.e.*, monomials that can be obtained using the bilinear map operation (this is a slight abuse of notation as a monomial m may not be deducible itself while its exponentiation g_2^m is deducible). A monomial $x_1x_2x_3$ from $\mathsf{mon}\,(t)$ is in $dm(t)$ if one or more of the following conditions hold:

- x_1, x_2 and x_3 are deducible from t,
- x_1, x_2 and $g_1^{x_3}$ are deducible from t,
- x_1, $g_1^{x_2}$ and $g_1^{x_3}$ are deducible from t.

We can now formalize the definition. Let t_2 and t_1 be two terms. A bijection σ from $\mathsf{pol}\,(t_2)$ to $\mathsf{pol}\,(t_1)$ is *linear relation preserving* for t_2 and t_1 if the same linear relations are verified between polynomials from t_2 and their image using σ. However monomials from $dm(t_2)$ cannot be renamed as they are linked to other parts of term t_2 due to the bilinear pairing. Formally, σ has to verify the following condition:

$$\forall p_1, ..., p_n \in \mathsf{pol}\,(t_2),\ \forall a_1, ..., a_n \in \mathbb{Z}, \quad \forall m_1, ..., m_{n'} \in dm(t_2),\ \forall b_1, ..., b_{n'} \in \mathbb{Z},$$
$$\sum_{i=1}^{n} a_i p_i = \sum_{j=1}^{n'} b_j m_j \Leftrightarrow \sum_{i=1}^{n} a_i (p_i \sigma) = \sum_{j=1}^{n'} b_j m_j$$

Reconsider our first example: t_1 is the term $(x_1, x_2, g_1^{x_3}, g_2^{x_4x_5x_6}, g_2^{x_1x_2x_3+x_4x_5x_6})$ and t_2 is the term $(x_1, x_2, g_1^{x_3}, g_2^{x_4x_5x_6}, g_2^{x_7x_8x_9})$. We define the bijection $\sigma = \{x_7x_8x_9 \mapsto x_1x_2x_3 + x_4x_5x_6\}$. We have that σ is *not* a linear relation preserving bijection for t_2 and t_1 because $x_1x_2x_3$ is in $dm(t_2)$ and

$$\begin{array}{ll} & (x_7x_8x_9) + (-1)(x_4x_5x_6) \neq x_1x_2x_3 \\ \text{but} & (x_7x_8x_9)\sigma + (-1)(x_4x_5x_6)\sigma = (x_1x_2x_3 + x_4x_5x_6) - (x_4x_5x_6) = x_1x_2x_3 \end{array}$$

Definition 3.4 *Two terms t_1 and t_2 are* equivalent up to renaming, *$t_1 \cong t_2$ if they are equivalent up to some renaming of keys of polynomials.*

$$\begin{array}{lll} t_1 \cong t_2 & \textit{iff} & \exists\sigma_1 \textit{ a renaming of } \mathbf{Keys} \\ & & \exists\sigma_2 \textit{ a bijection preserving linear relations from } \mathsf{pol}\,(t_2) \textit{ to } \mathsf{pol}\,(t_1) \\ & & \textit{such that } t_1 \equiv t_2\sigma_1\sigma_2 \end{array}$$

In this definition of equivalence, we have not considered renaming of **Exponents** to preserve simplicity but this can easily be added. Using this new definition, an interesting result is the decidability of equivalence up to renaming.

Proposition 3.5 *Let t_1 and t_2 be two terms. Equivalence up to renaming of t_1 and t_2 is decidable.*

Proof. As detailed in the proof of proposition 3.3, there exists an algorithm that takes as input a term t and outputs the finite set $K(t)$. This allows us to build an algorithm that takes as input a term t and outputs $\mathsf{pat}\,(t, K(t))$.

Let t_1 and t_2 be two terms. Then it is possible to compute $\mathsf{pat}\,(t_1, K(t_1))$ and $\mathsf{pat}\,(t_2, K(t_2))$ (and so equivalence *without* renaming, $\equiv$, is decidable).

In order to decide equivalence up to renaming of terms t_1 and t_2, we apply a unification algorithm recursively on $\mathsf{pat}\,(t_1, K(t_1))$ and $\mathsf{pat}\,(t_2, K(t_2))$ resulting in a renaming σ_1 and a bijection σ_2 from $\mathsf{pol}\,(t_2)$ to $\mathsf{pol}\,(t_1)$. This unification algorithm takes two terms u_1 and u_2 as an input and works as follows:

1. If u_1 is a pair (v_1, w_1) and u_2 is a pair (v_2, w_2) the algorithm is applied recursively on v_1 and v_2 resulting in σ_1 and σ_2. This algorithm is also applied recursively on w_1 and w_2 resulting in σ_1' and σ_2'. If σ_1 and σ_1' are compatible (*i.e.*, for any atomic key k that is in the domain of both σ_1 and σ_1', $k\sigma_1 = k\sigma_1'$) and σ_2 and σ_2' are also compatible, then u_1 and u_2 can be unified resulting in $\sigma_1 \cup \sigma_1'$ and $\sigma_2 \cup \sigma_2'$. Otherwise u_1 and u_2 cannot be unified and t_1 and t_2 are not equivalent up to renaming.

2. If u_1 is an encryption $\{v_1\}_{key_1}$ and u_2 is an encryption $\{v_2\}_{key_2}$ we proceed as for pairs in the previous point: v_1 and v_2 are unified, key_1 and key_2 are unified and the compatibility is checked.

3. If u_1 is an atomic key k_1 and u_2 is an atomic key k_2. Then $\sigma_1 = \{k_2 \mapsto k_1\}$ and $\sigma_2 = \emptyset$.

4. If u_1 is a key $g_2^{p_1}$ and u_2 is a key $g_2^{p_2}$ then $\sigma_1 = \emptyset$ and $\sigma_2 = \{p_2 \mapsto p_1\}$.

5. If u_1 is an exponentiation $g_1^{x_1}$ and u_2 is an exponentiation $g_1^{x_2}$ or if u_1 is an exponent x_1 and u_2 is an exponent x_2 and x_1 is equal to x_2, then u_1 and u_2 can be unified resulting in $\sigma_1 = \sigma_2 = \emptyset$. Otherwise t_1 and t_2 are not equivalent up to renaming.

6. Otherwise, u_1 and u_2 cannot be unified and terms t_1 and t_2 are not equivalent up to renaming.

Now, it only remains to check that σ_2 is a linear relation preserving bijection for t_2 and t_1. First the set $dm(t_2)$ is computed. Notice that elements of $dm(t_2)$ are monomials using exponents from t_2. For each possible monomial m, m is in $dm(t_2)$ if and only if $m = x_1x_2x_3$ and one of the three following holds:

- x_1, x_2 and x_3 are reachable in t_2 using $K(t_2)$.
- x_1, x_2 and $g_1^{x_3}$ are reachable in t_2 using $K(t_2)$.
- x_1, $g_1^{x_2}$ and $g_1^{x_3}$ are reachable in t_2 using $K(t_2)$.

In order to check that σ_2 preserves linear relations of t_2 we need to check that σ_2 does neither remove nor add any linear relation. To check whether σ_2 removes a linear relation in t_2 we use the following algorithm. Let P be an initially empty set of polynomials. The algorithm iterates on polynomials from $\mathsf{pol}\,(t_2)$. For each such polynomial p, the algorithm tests whether p is involved in a linear relation with polynomials from P and monomials from $dm(t_2)$. This can be tested by checking whether the system of linear equations $\sum_{1\leq i\leq n} \lambda_i p_i + \sum_{1\leq i\leq j} \lambda'_j m_j - p = 0$ with $P = \{p_1, \ldots, p_n\}$ and $dm(t_2) = \{m_1, \ldots, m_j\}$ has a solution, e.g. using Gauss elimination. If this is the case, then if $p\sigma_2$ verifies the same relation with $P\sigma_2$ and $dm(t_2)$, the algorithm continues, else if the relation is not satisfied by $p\sigma_2$, $P\sigma_2$ and $dm(t_2)$, then σ_2 is not linear relation preserving. If p is not involved in a linear combination with polynomials from P, then p is added to P. After that, the loop continues. As $\mathsf{pol}\,(t_2)$ is finite, this algorithm always terminates. To check whether σ_2 adds a linear relation to t_2, we use the previous algorithm and (equivalently) check whether σ_2^{-1} removes a linear relation in t_1. □

3.3 Examples

Here we give some examples that illustrate the choices we made when defining the equivalence. These choices are motivated by the possibilities of adversaries in the computational setting. Unlike [3], our symbolic model does not include symbolic constants like 0 or 1 as data. However these constants can be easily encoded using for instance two key names k_0 and k_1 which are explicitly revealed. Then 1 denotes k_1 and 0 denotes k_0. Instead of verifying the equivalence between t and t', we check whether (k_0, k_1, t) and (k_0, k_1, t') are equivalent.

1. $\{0\}_k \cong \{1\}_k$. This example shows that symmetric encryption perfectly hides its plaintext.

2. $(\{0\}_k, \{0\}_k) \cong (\{0\}_k, \{1\}_k)$. Symmetric encryption also hides equalities among the underlying plaintexts. To achieve this, encryption has to be probabilistic. As modular exponentiation is deterministic, we cannot ask modular exponentiation to hide such relations.

3. $(g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, g_2^{x_1x_2x_3}) \cong (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, g_2^{x'_1x'_2x'_3})$. This example illustrates security of Joux's protocol [29] against passive adversaries. The adversary observes the unfolding of the protocol where three exponentiations are exchanged. These exponentiations allows the three participants to build a shared secret key $g_2^{x_1x_2x_3}$. Then the adversary cannot distinguish the shared key from a randomly sampled element of the second group $g_2^{x'_1x'_2x'_3}$ (as the order of the group is prime, $g_2^{x'_1x'_2x'_3}$ has a uniform distribution over elements of the second group).

   Moreover the symbolic setting can be used to verify that each participant is able to compute the shared key. For example the first participant generates exponent x_1 and receives $g_1^{x_2}$ and $g_1^{x_3}$ from the second and third participants. Using this knowledge, he is able to compute the shared secret key as $x_1, g_1^{x_2}, g_1^{x_3} \vdash g_2^{x_1x_2x_3}$.

4. $(g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, \{0\}_{g_2^{x_1x_2x_3}}) \cong (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, \{1\}_{g_2^{x_1x_2x_3}})$. This example combines the Joux protocol with an exchange of secret information using the shared key. Thus in this example symmetric encryption and bilinear pairing are used simultaneously.

5. $(g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, x_4, x_5, x_6, g_2^{x_1x_2x_3+x_4x_5x_6}) \cong (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, x_4, x_5, x_6, g_2^{x'_1x'_2x'_3})$. Let t_2 be the second term in the equivalence relation. This example illustrates a more complex renaming. The adversary has access to some exponents from the key $g_2^{x_1x_2x_3+x_4x_5x_6}$ but is still unable to distinguish it from a randomly sampled key. $x_4x_5x_6$ can be seen as a vulnerable part of the key but $x_1x_2x_3$ makes the whole key secure. The two terms are equivalent up to renaming because bijection $\{x'_1x'_2x'_3 \mapsto x_1x_2x_3 + x_4x_5x_6\}$ is linear relation preserving; indeed $x'_1x'_2x'_3$ and $x_1x_2x_3 + x_4x_5x_6$ are both not involved in any linear relation with monomials from $dm(t_2)$.

6. In the following example, there are two shared keys.

$$\begin{array}{ll} & (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, x_4, x_5, x_6, g_2^{x_1x_2x_3+x_4x_5x_6}, g_2^{x_1x_2x_3}) \\ \not\cong & (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, x_4, x_5, x_6, g_2^{x'_1x'_2x'_3}, g_2^{x'_4x'_5x'_6}) \end{array}$$

Let t_1 and t_2 be the first and second term in this (non-)equivalence relation. The bijection $\sigma = \{x'_1x'_2x'_3 \mapsto x_1x_2x_3 + x_4x_5x_6 \ ; \ x'_4x'_5x'_6 \mapsto x_1x_2x_3\}$ is not linear relation preserving. Indeed, the monomial $x_4x_5x_6$ is in $dm(t_2)$ and there is a relation among polynomials used in the two keys of t_1 and $x_4x_5x_6$ which is not true in t_2:

$$\begin{array}{rcl} (x'_1x'_2x'_3)\sigma + (-1)(x'_4x'_5x'_6)\sigma & = & x_4x_5x_6 \\ (x'_1x'_2x'_3) + (-1)(x'_4x'_5x'_6) & \neq & x_4x_5x_6 \end{array}$$

4 The Computational Setting

In this section, we formalize the mapping between symbolic terms and distributions of bit-strings. This mapping depends on the algorithms used to implement the two cryptographic primitives used in the symbolic setting: symmetric encryption and pairing.

4.1 Encryption Scheme

We recall the standard definition for symmetric encryption schemes. A symmetric encryption scheme $\mathcal{SE}$ is defined by three algorithms $\mathcal{KG}$, $\mathcal{E}$ and $\mathcal{D}$. The key generation algorithm $\mathcal{KG}$ takes as input the security parameter η and outputs a key k. The encryption algorithm $\mathcal{E}$ is randomized. It takes as input a bit-string s and a key k and returns the encryption of s using k. The decryption algorithm $\mathcal{D}$ takes as input a bit-string c representing a ciphertext and a key k and outputs the corresponding plaintext. Given $k \leftarrow \mathcal{KG}(\eta)$, we have that for any bit-string s, if $c \leftarrow \mathcal{E}(s, k)$ then it is required that $\mathcal{D}(c) = s$.

In order to characterize security of a symmetric encryption scheme, we use the classical IND-CPA (indistinguishability against chosen plaintext attacks) notion [23].

IND-CPA security. In this paper we use schemes that satisfy length-concealing semantic security[1]. 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 bit-strings, an important point is that bs_0 and bs_1 may have different lengths. The oracle returns ciphertext $\mathcal{E}(bs_b, k)$. The goal of the adversary $\mathcal{A}$ is to guess the value of bit b and for that purpose $\mathcal{A}$ has access to oracle $LR^b_{\mathcal{SE}}$. His advantage is defined as the probability that he outputs 1 when using oracle $LR^1_{\mathcal{SE}}$ minus the probability that he outputs 1 when using oracle $LR^0_{\mathcal{SE}}$.

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

An encryption scheme $\mathcal{SE}$ is said to be IND-CPA secure if the advantage of any polynomial-time adversary $\mathcal{A}$ is negligible in η.

[1] Such schemes can only exist if the maximum length of plaintexts is bounded, however we do not take this into account in this paper.

The difference with the standard notion of semantic security is that an adversary can call oracle $LR^b_{\mathcal{SE}}$ on two bit-strings bs_0 and bs_1 of different lengths. Therefore in order to be secure for our notion, an encryption scheme has to hide the length of the plaintext. By abuse of notation we call the resulting scheme also $\mathsf{IND\text{-}CPA}$ secure.

4.2 Pairing

A *bilinear pairing instance generator* is defined as a probabilistic polynomial-time algorithm IG which given a security parameter η outputs a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, g_1, e)$ composed of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, a generator g_1 of $\mathbb{G}_1$ and a cryptographic bilinear map e between $\mathbb{G}_1$ and $\mathbb{G}_2$. A generator g_2 of group $\mathbb{G}_2$ is obtained by applying e to (g_1, g_1).

BDDH security. An instance generator IG satisfies the *Bilinear Decisional Diffie-Hellman assumption*, BDDH, iff for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against BDDH, $\mathrm{Adv}^{\mathsf{BDDH}}_{\mathcal{A},IG}$, defined below is negligible in η.

$$\begin{aligned}\mathrm{Adv}^{\mathsf{BDDH}}_{\mathcal{A},IG}(\eta) \quad = \quad & \mathbb{P}\left[\begin{array}{c}(q, \mathbb{G}_1, \mathbb{G}_2, g_1, e) \leftarrow IG(\eta)\\ x, y, z \leftarrow \mathbb{Z}_q\end{array}, \mathcal{A}(g_1, g_1^x, g_1^y, g_1^z, g_2^{xyz}) = 1\right] \\ & -\mathbb{P}\left[\begin{array}{c}(q, \mathbb{G}_1, \mathbb{G}_2, g_1, e) \leftarrow IG(\eta)\\ x, y, z, r \leftarrow \mathbb{Z}_q\end{array}, \mathcal{A}(g_1, g_1^x, g_1^y, g_1^z, g_2^r) = 1\right]\end{aligned}$$

This means that an adversary that is given g_1^x, g_1^y and g_1^z can only make the difference between g_2^{xyz} and a random group element with negligible probability.

4.3 Computational Semantics of Terms

Computational semantics depend on a symmetric encryption scheme $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ and of an instance generator IG. In order to transform elements of the second group into keys usable for $\mathcal{SE}$, we assume the existence of a key extractor [17] algorithm Kex (this can be done for example by extracting randomness using an entropy smoothing hash function [27]). We suppose that the distribution of keys generated by $\mathcal{KG}$ is equal to the distribution obtained by applying Kex to a random element of $\mathbb{G}_2$ (which is the second group generated by IG). We associate to each symbolic term t a distribution of bit-strings $[\![t]\!]_{\mathcal{SE},IG}$ that depends on the security parameter η. This distribution is defined by the following random algorithm:

1. Algorithm IG is used to generate two paired groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q and of generators $\widehat{g_1}$ and $\widehat{g_2}$. For each key k from t, a value $\widehat{k}$ is randomly drawn using $\mathcal{KG}$. For each exponent x, a value $\widehat{x}$ is randomly sampled in $\mathbb{Z}_q$ equipped with the uniform distribution.

2. Then the bit-string evaluation of term t is computed recursively on the structure of t:

- If t is a key k or an exponent x, then the value $\widehat{t}$ is returned.
- If t is an exponentiation g_1^x, then the exponentiation of $\widehat{g_1}$ to the power of $\widehat{x}$ is returned.
- If t is an exponentiation g_2^p, then the algorithm computes the value n of p in $\mathbb{Z}_q$, and the exponentiation of $\widehat{g_2}$ to the power of n is returned.
- If t is a pair (t_1, t_2), the algorithm is applied recursively on t_1 holding bs_1 and on t_2 holding bs_2. The output of the algorithm is the concatenation of bs_1 and bs_2.
- If t is an encryption $\{t'\}_k$, the algorithm is applied recursively on t' holding bs' and on k holding bs_k. The output of the algorithm is $\mathcal{E}(bs', bs_k)$.
- If t is an encryption $\{t'\}_{g_2^p}$, the algorithm is applied recursively on t' holding bs' and on g_2^p holding bs_k. The output of the algorithm is $\mathcal{E}\left(bs', \mathsf{Kex}(bs_k)\right)$.

5 Soundness of the Symbolic Model

In this section we prove that the extension of the Abadi-Rogaway logic given in section 3 is computationally sound when implemented using an IND-CPA encryption scheme and using an instance generator satisfying BDDH: if two terms are equivalent up to renaming in the symbolic setting, their evaluations (given by the computational semantics of section 4) are computationally indistinguishable.

Well-formed Terms. Our soundness result is only true for terms that make a correct use of the bilinear pairing. Such terms are called *well-formed* terms. Formally a term t is *well-formed* if for any monomial m in $\mathsf{mon}\,(t)$:

- either for any monomial m' in $\mathsf{mon}\,(t)$ different from m, m and m' do not have any common exponent;
- or none of the three exponents used by m occurs as data in t.

This technical restriction is necessary to obtain soundness. Indeed let us consider $t_1 = (x, y, g_2^{xz_1z_2}, g_2^{yz_1z_2})$ and $t_2 = (x, y, g_2^{r_1r_2r_3}, g_2^{r_4r_5r_6})$. Note that t_1 is not well-formed as xz_1z_2 and yz_1z_2 have common exponents (z_1 and z_2) and exponents x and y occur as data. Terms t_1 and t_2 are equivalent up to renaming. However it is possible to build an adversary $\mathcal{A}$ that can distinguish the corresponding distributions efficiently (the precise definition of indistinguishability will be given below). Adversary $\mathcal{A}$ takes as input (x, y, U, V) and has to decide whether $U = g_2^{xz_1z_2}$ and $V = g_2^{yz_1z_2}$ or $U = g_2^{r_1r_2r_3}$ and $V = g_2^{r_4r_5r_6}$. $\mathcal{A}$ proceeds as follows:

- compute x^{-1} and y^{-1};

- output 1 if $U^{x^{-1}} = V^{y^{-1}}$;
- output 0 otherwise.

If $\mathcal{A}$ outputs 1 it was indeed given the distribution corresponding to t_1 with probability close to 1 (the probability that $r_1r_2r_3x^{-1} = r_4r_5r_6y^{-1}$ is negligible). Otherwise, if A outputs 0 it must have been given the distribution corresponding to t_2. Hence, $\mathcal{A}$ efficiently distinguishes two equivalent terms. We forbid such use of bilinear pairing by considering only well-formed terms.

Acyclicity Restrictions. The importance of key cycles was already described in [3]. In the setting of [3] 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 general IND-CPA is not sufficient to prove any soundness result in presence of key cycles. 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:

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

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). Hence, as in numerous previous work we forbid the symbolic terms to contain such cycles. (Another possibility to handle key cycles is to consider stronger computational requirements like Key Dependent Message – KDM – security as done in [4].)

We now define a similar notion of key cycles in our setting. For any term t, let $kp(t)$ be the set of polynomials p such that g_2^p occurs at a key position in t and g_2^p is not deducible from t. Let $pm(t)$ be the set of monomials $x_1x_2x_3$ such that either:

- x_1, x_2 and x_3 occur as data in t;
- x_1 and x_2 occur as data in t and $g_1^{x_3}$ also appears in t;
- x_1 occurs as data in t and $g_1^{x_2}$ and $g_1^{x_3}$ also appear in t.

A term t is *acyclic* if the two following restrictions are verified.

- For any p in $kp(t)$, p is linearly independent from any other polynomials from $\mathsf{pol}\,(t)$ and from monomials from $pm(t)$, *i.e.*, if $\mathsf{pol}\,(t) = \{p, p_1, ..., p_n\}$ and $pm(t) = \{m_1, ..., m_{n'}\}$ then there does not exist any integers a, a_1 to a_n and b_1 to $b_{n'}$ such that $a \neq 0$ and:

$$a.p = \sum_{i=1}^{n} a_i p_i + \sum_{j=1}^{n'} b_j m_j$$

- There exists an order $\prec$ among keys used in t such that for any subterm $\{u\}_{key}$ of t, either key is deducible from t or for each key key' that occurs in u, $key' \prec key$.

We illustrate the notion of key cycles on several examples.

- The terms $\{k\}_k$ and $(\{k_1\}_{k_2}, \{k_2\}_{k_1})$ contain key cycles, as those considered already in [3].
- The term $t = (\{k\}_{g_2^{x_1 x_2 x_3}}, \{g_2^{x_1 x_2 x_3}\}_k)$ obviously contains a key cycle while $(\{k\}_{g_2^{x_1 x_2 x_3}}, \{g_2^{x_1 x_2 x_3}\}_{k'})$ does not.
- The term $t = (g_1^{x_1}, g_1^{x_2}, x_3, \{k\}_{g_2^{x_1 x_2 x_3}}, \{g_2^{x_1 x_2 x_3}\}_k)$ is acyclic as $g_2^{x_1 x_2 x_3}$ is deducible (and hence $kp(t) = \emptyset$).
- The term $t = \{(x_1, g_1^{x_2}, g_1^{x_3})\}_{g_2^{2 x_1 x_2 x_3}}$ contains a key cycle because $pm(t) = \{x_1 x_2 x_3\}$ and $2x_1 x_2 x_3 \in kp(t)$ is linearly dependent.

Our acyclicity restriction is stronger than what is strictly required for computational soundness: for example $\{\{k\}_{k'}\}_k$ is considered as a cycle whereas it is not problematic as the underlying k is hidden by k'. We consider this stronger definition of acyclicity as it is easier to define and it also makes our main proof simpler.

5.1 Soundness Result

Indistinguishable Distributions. Before giving our soundness result, we recall the usual notion of indistinguishable distributions. Intuitively, two distributions D_1 and D_2 are computationally indistinguishable if for any adversary $\mathcal{A}$, the probability for $\mathcal{A}$ to detect the difference between a randomly sampled element of D_1 and a randomly sampled element of D_2 is negligible in η.

Definition 5.1 *Let D_1 and D_2 be two distributions (that depend on η). The advantage of an adversary $\mathcal{A}$ in distinguishing D_1 and D_2 is defined by:*

$$\mathrm{Adv}_{\mathcal{A}}^{D_1,D_2} = \mathbb{P}\left[x \leftarrow D_1(\eta)\ ;\ \mathcal{A}(x) = 1\right] - \mathbb{P}\left[x \leftarrow D_2(\eta)\ ;\ \mathcal{A}(x) = 1\right]$$

Distributions D_1 and D_2 are computationally indistinguishable*, written $D_1 \approx D_2$, if the advantage for any adversary $\mathcal{A}$ in distinguishing D_1 and D_2 is negligible.*

Then our main soundness result states that distributions related to equivalent terms are computationally indistinguishable.

Proposition 5.2 *Let t_0 and t_1 be two acyclic well-formed terms, such that $t_0 \cong t_1$. Let $\mathcal{SE}$ be a symmetric encryption scheme that is secure for* $\mathsf{IND\text{-}CPA}$ *and IG be an instance generator satisfying* BDDH, *then $[\![t_0]\!]_{\mathcal{SE},IG} \approx [\![t_1]\!]_{\mathcal{SE},IG}$.*

Proof for proposition 5.2

In order to prove our main soundness result, we introduce some intermediate lemmas. First we prove that BDDH implies an extended version of BDDH. Intuitively this first lemma states that if BDDH holds and $\mathcal{A}$ is an adversary that is given some exponents x_1 to x_α and some exponentiations $g_1^{y_1}$ to $g_1^{y_\beta}$, $\mathcal{A}$ cannot distinguish exponentiations of linearly independent polynomials $g_2^{p_1}$ to $g_2^{p_\gamma}$ from exponentiations of fresh exponents $g_2^{r_{1,1}r_{1,2}r_{1,3}}$ to $g_2^{r_{\gamma,1}r_{\gamma,2}r_{\gamma,3}}$.

Lemma 5.3 *Let $X = (x_i)_{1\leq i\leq\alpha}$ and $Y = (y_i)_{1\leq i\leq\beta}$ be $\alpha+\beta$ exponents. Let $P = (p_i)_{1\leq i\leq\gamma}$ be γ polynomials such that there are no linear relations between the p_i and the set of monomials $\{xyz,\ x,y,z \in X\} \cup \{xyz,\ x,y \in X,\ z \in Y\} \cup \{xyz,\ x \in X,\ y,z \in Y\}$.*

If IG is an instance generator satisfying BDDH *and the two following terms are well-formed then:*

$$[\![x_1,...,x_\alpha,g_1^{y_1},...,g_1^{y_\beta},g_2^{p_1},...,g_2^{p_\gamma}]\!]_{IG} \approx [\![x_1,...,x_\alpha,g_1^{y_1},...,g_1^{y_\beta},g_2^{q_1},...,g_2^{q_\gamma}]\!]_{IG}$$

where each q_i is a product of three fresh exponents $r_{i,1}r_{i,2}r_{i,3}$, i.e., *the part of the distribution related to $g_2^{q_i}$ corresponds to a random group element.*

Proof. First, note that as the order q of the group $\mathbb{G}_2$ is prime, in the computational setting $g_2^{Z_1Z_2}$ and $g_2^{Z_3}$ have the same distribution (Z_1, Z_2, and Z_3 are three independent random variables uniformly sampled over $\mathbb{Z}_q$).

The proof of this lemma is done in two steps.

- The first step consists in replacing monomials in the p_i that are not in $dm(x_1,...,x_\alpha,g_1^{y_1},...,g_1^{y_\beta})$ with fresh monomials. This results in a new term whose computational distribution is indistinguishable from the original term distribution.
- Then, in the second step we prove that the computational distribution of this new term using fresh monomials is exactly equal to the distribution related to $x_1,\ldots,x_\alpha,g_1^{y_1},\ldots,g_1^{y_\beta},g_2^{q_1},\ldots,g_2^{q_\gamma}$.

Step 1. Let M be the set of monomials from P that are not in $\{xyz,\ x,y,z \in X\} \cup \{xyz,\ x,y \in X,\ z \in Y\} \cup \{xyz,\ x \in X,\ y,z \in Y\}$. The first step consists in using BDDH to replace these monomials with fresh monomials $r_1r_2r_3$. Let p'_1 to p'_γ be polynomials p_1 to p_γ where each monomial of M has been replaced with a fresh monomial. We prove that:

$$[\![x_1,\ldots,x_\alpha,g_1^{y_1},\ldots,g_1^{y_\beta},g_2^{p_1},\ldots,g_2^{p_\gamma}]\!]_{IG} \approx [\![x_1,\ldots,x_\alpha,g_1^{y_1},\ldots,g_1^{y_\beta},g_2^{p'_1},\ldots,g_2^{p'_\gamma}]\!]_{IG}$$

This proof is done by induction on the number j of monomials in M that use at least one exponent which is also present in X, Y or in any other monomial used in a polynomial from P.

If $j = 0$ then for each monomial m used in p_1 to p_γ, m uses exponents that are not in X or Y nor in any other monomial from polynomials of P. Thus p_1 to p_γ are equal to p'_1 to p'_γ up to renaming of the exponents and so:

$$[\![x_1, \dots, x_\alpha, g_1^{y_1}, \dots, g_1^{y_\beta}, g_2^{p_1}, \dots, g_2^{p_\gamma}]\!]_{IG} = [\![x_1, \dots, x_\alpha, g_1^{y_1}, \dots, g_1^{y_\beta}, g_2^{p'_1}, \dots, g_2^{p'_\gamma}]\!]_{IG}$$

If $j > 0$ then let $m = xyz$ be a monomial in M that uses an exponent from X or Y or from another monomial of P and let m' be a fresh monomial. Let p''_1 to p''_γ be polynomials p_1 to p_γ where m has been replaced with m'. There are two cases to consider:

- First if x, y and z do not appear in X. Let $\mathcal{A}$ be an adversary trying to distinguish distribution

  $$[\![x_1, \dots, x_\alpha, g_1^{y_1}, \dots, g_1^{y_\beta}, g_2^{p_1}, \dots, g_2^{p_\gamma}]\!]_{IG}$$

  from distribution

  $$[\![x_1, \dots, x_\alpha, g_1^{y_1}, \dots, g_1^{y_\beta}, g_2^{p''_1}, \dots, g_2^{p''_\gamma}]\!]_{IG}.$$

  We build an adversary $\mathcal{B}$ against BDDH that executes $\mathcal{A}$ as a subroutine. As $\mathcal{B}$ tries to break BDDH, $\mathcal{B}$ receives four arguments (A, B, C, D). Intuitively, $\mathcal{B}$ uses the inputs A, B, C and D for g_1^x, g_1^y, g_1^z and g_2^{xyz}. $\mathcal{B}$ queries $\mathcal{A}$ with the input

  $$a_1, \dots, a_\alpha, b_1, \dots, b_\beta, c_1, \dots, c_\gamma$$

  where

  - a_i are values in $\mathbb{Z}_q$ randomly generated by $\mathcal{B}$;
  - b_i is computed as follows. If y_i equals x, y or z then b_i is set to A, B or C respectively. Otherwise b_i is set to g_1^u where $u = a_j$ if $y_i = x_j$ for some j or u is randomly sampled from $\mathbb{Z}_q$;
  - for each monomial m_0 appearing in p_i $\mathcal{B}$ computes the implementation for $g_2^{m_0}$ as follows. If $m_0 = m$ then $g_2^{m_0}$ is implemented by D. If m_0 shares two exponents with m, for example $m_0 = xyz'$, then the corresponding value is generated using the bilinear map: $\mathcal{B}$ computes $e(A, B)^c$ where c is either freshly generated by $\mathcal{B}$ or has been previously generated for z'. If m_0 only shares one exponent with m, for example $m_0 = xy'z'$, then $\mathcal{B}$ computes A^{bc} where where b and c are either freshly generated by $\mathcal{B}$ or have been previously genrated for y' and z'. Given the implementations of $g_2^{m_0}$ for each m_0 in p_i $\mathcal{B}$ computes implementations for each $g_2^{p_i}$ and use these values for $c_1, \dots, c_\gamma$.

Finally, $\mathcal{B}$ returns the same output as $\mathcal{A}$. The advantage of $\mathcal{B}$ against BDDH is given by

$$\mathrm{Adv}^{\mathsf{BDDH}}_{\mathcal{B},IG}(\eta) = \mathbb{P}\left[\mathcal{B}(g_1^x, g_1^y, g_1^z, g_2^{xyz}) = 1\right] - \mathbb{P}\left[\mathcal{B}(g_1^x, g_1^y, g_1^z, g_2^r) = 1\right]$$

When $\mathcal{B}$ receives as input $(g_1^x, g_1^y, g_1^z, g_2^{xyz})$, $\mathcal{A}$ is given a sample from distribution $[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}]\!]_{IG}$.

When $\mathcal{B}$ receives as input $(g_1^x, g_1^y, g_1^z, g_2^r)$, $\mathcal{A}$ is given a sample from distribution $[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p''_1}, \ldots, g_2^{p''_\gamma}]\!]_{IG}$.

Therefore the advantage of $\mathcal{A}$ in distinguishing the two distributions is equal to the advantage of $\mathcal{B}$ against BDDH. As BDDH holds, the advantage of $\mathcal{B}$ is negligible and so the advantage of $\mathcal{A}$ is also negligible. Hence,

$$\begin{gathered}[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}]\!]_{IG} \\ \approx \\ [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p''_1}, \ldots, g_2^{p''_\gamma}]\!]_{IG}\end{gathered}$$

- If x appears in X, then by definition of M either y or z does not appear in X and Y. Let us suppose that it is y. Exponent y only appears in m and, as the order of the group is prime, we have the following equality between distributions:

  $$[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{m}]\!]_{IG} = [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{m'}]\!]_{IG}$$

  Moreover as the original terms are well-formed and x occurs as data, m does not share any exponent with other monomials used in P. Hence, we also obtain that:

  $$\begin{gathered}[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}]\!]_{IG} \\ \approx \\ [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p''_1}, \ldots, g_2^{p''_\gamma}]\!]_{IG}\end{gathered}$$

We have proved that:

$$[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}]\!]_{IG} \approx [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p''_1}, \ldots, g_2^{p''_\gamma}]\!]_{IG}$$

where p''_1 to p''_n use $j-1$ monomials that use an exponent from $X \cup Y$. Hence using our induction hypothesis, we get that

$$[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p''_1}, \ldots, g_2^{p''_\gamma}]\!]_{IG} \approx [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p'_1}, \ldots, g_2^{p'_\gamma}]\!]_{IG}$$

And so we proved that

$$[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}]\!]_{IG} \approx [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p'_1}, \ldots, g_2^{p'_\gamma}]\!]_{IG}$$

Step 2. Let n be the number of elements in M.

For the second step, we recall that (as q is prime), the number of solutions over $\mathbb{Z}_q$ for a linear system of γ independent equations involving n variables is $q^{n-\gamma}$.

Let $(\hat{x}_i)_{1\leq i\leq\alpha}$, $(\hat{y}_i)_{1\leq i\leq\beta}$ and $(\hat{p}_i)_{1\leq i\leq\gamma}$ be elements of $\mathbb{Z}_q$. We now compute the probability for the distribution to output value v defined by:

$$v = \left(\hat{x}_1, \ldots, \hat{x}_\alpha, g_1^{\hat{y}_1}, \ldots, g_1^{\hat{y}_\beta}, g_2^{\hat{p}_1}, \ldots, g_2^{\hat{p}_\gamma}\right)$$

It is important to see that this is a computational value and not a symbolic term.

Then we associate to each monomial from M a variable over $\mathbb{Z}_q$ and we obtain a system involving γ linear equations using n variables.

$$\left(\hat{x}_1, \ldots, \hat{x}_\alpha, g_1^{\hat{y}_1}, \ldots, g_1^{\hat{y}_\beta}, g_2^{p'_1}, \ldots, g_2^{p'_\gamma}\right) = v$$

(The system is given by the equations between $g_2^{\cdot}$ as the other equalities are trivially satisfied.) The number of solutions of this system is $q^{n-\gamma}$. Hence when randomly sampling values for monomials in M, the probability to obtain v is $q^{n-\gamma}/q^n$ which is equal to $q^{-\gamma}$.

On the other side, the probability to obtain v by randomly sampling γ group elements for the $g_2^{q_i}$ is also equal to $q^{-\gamma}$ so the distributions are identical:

$$[\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p'_1}, \ldots, g_2^{p'_\gamma}]\!]_{IG} = [\![x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{q_1}, \ldots, g_2^{q_\gamma}]\!]_{IG}$$

And so we obtain the expected result. □

In order to introduce the following lemmas, we define the computational semantics of patterns (*i.e.*, terms using □) by extending the semantics for terms with $[\![\Box]\!]_{\mathcal{SE},IG} = 0$. Our second lemma states that evaluations of a term and of its pattern are indistinguishable in the computational setting.

Lemma 5.4 *Let t be an acyclic well-formed term. Let $\mathcal{SE}$ be an* IND-CPA *secure symmetric encryption scheme and let IG be an instance generator satisfying* BDDH. *Then we have that*

$$[\![t]\!]_{\mathcal{SE},IG} \approx [\![\mathsf{pat}\,(t, K(t))]\!]_{\mathcal{SE},IG}$$

Proof. Let t be an acyclic well-formed term. Then any p in $kp(t)$ is linearly independent of any other polynomials from $\mathsf{pol}\,(t)$. Let $\overline{K(t)}$ be the set of keys and exponentiations g_2^p used at a key position in t that are not in $K(t)$, *i.e.*, that are not deducible. Let key be a metavariable over $\overline{K(t)}$. As t is acyclic there exists a total order $\prec$ between elements of $\overline{K(t)}$ such that for any subterm $\{t'\}_{key}$ of t, key' can only appear in t' if $key' \prec key$.

This proof follows the lines of the main proof in [3]. The main difference with the original proof is that keys can be an exponentiation g_2^p. However as p is not involved in any linear relation, using this key is indistinguishable from using an atomic key.

Let us now detail the proof. Let n be the number of keys in $\overline{K(t)}$ and let key_i be the i^{th} key from $\overline{K(t)}$ with respect to $\prec$, *i.e.*:

$$\overline{K(t)} = \{key_1, \ldots, key_n\} \text{ and } key_1 \prec key_2 \prec \ldots \prec key_n$$

For i in $[0, n]$, term t_i is defined as $pat_i(t)$ where pat_i is recursively defined by:

$$\begin{array}{lcll} pat_i((t_1, t_2)) & = & \big(pat_i(t_1), pat_i(t_2)\big) & \\ pat_i(\{t'\}_{key}) & = & \{\Box\}_{key} & \text{if } key = key_j \text{ for } j \leq i \\ pat_i(\{t'\}_{key}) & = & \{pat_i(t')\}_{key} & \text{else} \\ pat_i(a) & = & a & \text{for } a \text{ in } x,\ k,\ g_1^x \text{ and } g_2^p \end{array}$$

In t_i, encryptions using keys key_j for $j \leq i$ have been replaced by encryptions of $\Box$. Hence $pat_0(t) = t$ and $pat_n(t) = \mathsf{pat}\,(t, K(t))$. The advantage of an adversary $\mathcal{A}$ which tries to distinguish $[\![t]\!]_{\mathcal{SE},IG}$ and $[\![\mathsf{pat}\,(t, K(t))]\!]_{\mathcal{SE},IG}$ can be written as:

$$\begin{array}{ll} & \mathrm{Adv}_{\mathcal{A}}^{[\![t]\!]_{\mathcal{SE},IG},[\![\mathsf{pat}(t,K(t))]\!]_{\mathcal{SE},IG}} = \mathrm{Adv}_{\mathcal{A}}^{[\![t_0]\!]_{\mathcal{SE},IG},[\![t_n]\!]_{\mathcal{SE},IG}} \\ = & \mathbb{P}\left[x \leftarrow [\![t_0]\!]_{\mathcal{SE},IG}\ ;\ \mathcal{A}(x) = 1\right] - \mathbb{P}\left[x \leftarrow [\![t_n]\!]_{\mathcal{SE},IG}\ ;\ \mathcal{A}(x) = 1\right] \\ = & \displaystyle\sum_{i=1}^{n} \left(\mathbb{P}\left[x \leftarrow [\![t_{i-1}]\!]_{\mathcal{SE},IG}\ ;\ \mathcal{A}(x) = 1\right] - \mathbb{P}\left[x \leftarrow [\![t_i]\!]_{\mathcal{SE},IG}\ ;\ \mathcal{A}(x) = 1\right]\right) \\ = & \displaystyle\sum_{i=1}^{n} \mathrm{Adv}_{\mathcal{A}}^{[\![t_{i-1}]\!]_{\mathcal{SE},IG},[\![t_i]\!]_{\mathcal{SE},IG}} \end{array}$$

We build n adversaries $(\mathcal{B}_i)_{1 \leq i \leq n}$ against $\mathsf{IND\text{-}CPA}$ that use $\mathcal{A}$ as a subroutine and such that the advantage of $\mathcal{B}_i$ against $\mathsf{IND\text{-}CPA}$ can be linked to the advantage $\mathrm{Adv}_{\mathcal{A}}^{[\![t_{i-1}]\!]_{\mathcal{SE},IG},[\![t_i]\!]_{\mathcal{SE},IG}}$.

Each adversary $\mathcal{B}_i$ uses his challenge key for key key_i and has access to a left-right encryption oracle $LR^b_{\mathcal{SE}}$. If key key_i is an exponentiation g_2^p then as p is not involved in any linear relation and because of lemma 5.3, the evaluation g_2^p is indistinguishable from a random group element. The key extraction algorithm Kex applied to a random group element returns a random key (whose distribution corresponds to the one of $\mathcal{KG}$). Hence, using g_2^p is indistinguishable from using a fresh atomic key.

Adversary $\mathcal{B}_i$ generates values for each atom used in t. For any subterm a of t which is of the form x, k, g_1^x or g_2^p, $\mathcal{B}_i$ computes a bit-string value bs_a according to the values generated previously. Using his left-right encryption oracle, $\mathcal{B}_i$ computes a bit-string bs which is either an evaluation of t_i or an evaluation of t_{i-1} depending on the challenge bit b. Formally bit-string bs is obtained by applying the recursive $eval_i$ function on t_{i-1}:

$$\begin{array}{lcll} eval_i((t_1, t_2)) & = & eval_i(t_1) \cdot eval_i(t_2) & \\ eval_i(\{t'\}_{key_i}) & = & LR^b_{\mathcal{SE}}(0, eval_i(t')) & \\ eval_i(\{t'\}_{key}) & = & \mathcal{E}(eval_i(t'), eval_i(key)) & \text{for } key \neq key_i \\ eval_i(\Box) & = & 0 & \\ eval_i(a) & = & bs_a & \text{for } a \text{ in } x,\ k,\ g_1^x \text{ and } g_2^p \end{array}$$

This algorithm works well as due to acyclicity key_i does not occur as data in t_{i-1}. Note that oracle $LR^b_{\mathcal{SE}}$ can be given as arguments two bit-strings of different lengths. This is why we had to assume that encryption scheme $\mathcal{SE}$ is length-concealing.

After computing bs, $\mathcal{B}_i$ executes $\mathcal{A}$ with input bs and returns the same result as $\mathcal{A}$. Let us sum up how $\mathcal{B}_i$ works:

```
Adversary $\mathcal{B}_i^{LR^b_{\mathcal{SE}}}(\eta)$
    for each $a$, compute $bs_a$
    $bs \leftarrow eval_i(t_{i-1})$
    $d \leftarrow \mathcal{A}(bs)$
    return $d$
```

If bit b equals 0, then $\mathcal{A}$ is confronted to an evaluation of t_i whereas if b equals 1, then $\mathcal{A}$ is given an evaluation of t_{i-1}. The advantages of $\mathcal{A}$ and $\mathcal{B}_i$ can be linked in the following way:

$$\mathrm{Adv}_{\mathcal{A}}^{[\![t_{i-1}]\!]_{\mathcal{SE},IG},[\![t_i]\!]_{\mathcal{SE},IG}} = \mathrm{Adv}^{\mathsf{CPA}}_{\mathcal{SE},\mathcal{B}_i}$$

Therefore we have that:

$$\mathrm{Adv}_{\mathcal{A}}^{[\![t]\!]_{\mathcal{SE},IG},[\![\mathsf{pat}(t,K(t))]\!]_{\mathcal{SE},IG}} = \sum_{i=1}^{n} \mathrm{Adv}^{\mathsf{CPA}}_{\mathcal{SE},\mathcal{B}_i}$$

As $\mathcal{SE}$ is assumed to be $\mathsf{IND\text{-}CPA}$ secure, the advantage of $\mathcal{B}_i$ is negligible for any i. Hence the advantage of $\mathcal{A}$ is also negligible. □

Our third lemma states that two patterns equal up to renaming are also indistinguishable in the computational setting.

Lemma 5.5 *Let t_0 and t_1 be two well-formed terms such that* $\mathsf{pat}\,(t_0, K(t_0)) \cong \mathsf{pat}\,(t_1, K(t_1))$. *Let $\mathcal{SE}$ be a symmetric encryption scheme (not necessarily secure) and let IG be an instance generator satisfying* BDDH, *then*

$$[\![\mathsf{pat}\,(t_0, K(t_0))]\!]_{\mathcal{SE},IG} \approx [\![\mathsf{pat}\,(t_1, K(t_1))]\!]_{\mathcal{SE},IG}$$

Proof. Let t'_0 be the term $\mathsf{pat}\,(t_0, K(t_0))$ and t'_1 be the term $\mathsf{pat}\,(t_1, K(t_1))$. There exists a renaming of **Keys** σ_1 and a bijection σ_2 preserving linear relations between polynomials from t_1 to t_0 such that $t'_0 = t'_1\sigma_1\sigma_2$. Permutation of keys is easy to handle: $[\![t'_1\sigma_1]\!]_{\mathcal{SE},IG}$ and $[\![t'_1]\!]_{\mathcal{SE},IG}$ output exactly the same distribution.

There only remains to prove that $[\![t'_0]\!]_{\mathcal{SE},IG} \approx [\![t'_1\sigma_1]\!]_{\mathcal{SE},IG}$. For this purpose, let $u_0 = t'_0$ and $u_1 = t'_1\sigma_1$. Let $\mathcal{A}$ be an adversary trying to distinguish the distribution related to u_0 from the distribution related to u_1. In the remaining, we prove that the advantage of $\mathcal{A}$ is negligible if the BDDH assumption holds. For this purpose, we introduce a term u such that:

$$\mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}} = \mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u]\!]_{\mathcal{SE},IG}} + \mathrm{Adv}_{\mathcal{A}}^{[\![u]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}}$$

Intuitively u is equal to u_0 where polynomials have been replaced by fresh monomials whenever possible while conserving linear equalities. u is also equal to u_1 where the same modification has been applied. From there, due to the nature of u it is easy to prove that the two advantages on the right part are negligible using lemma 5.3.

First let us define the following sets:

1. Let $X = (x_i)_{1 \leq i \leq \alpha}$ be the exponents that are deducible from u_0 (using u_1 instead of u_0 would give exactly the same X as $u_0 = u_1 \sigma_2$).
2. Let $Y = (y_i)_{1 \leq i \leq \beta}$ be the exponents such that $g_1^{y_i}$ is deducible from u_0 (as previously, using u_1 instead of u_0 would give exactly the same Y).
3. Let $M = (m_i)_{1 \leq i \leq \delta}$ be the set of monomials $dm(u_0)$ which can easily be obtained from X and Y.
4. The two sets of polynomials $P_0 = (p_{0,i})_{1 \leq i \leq \gamma}$ and $P_1 = (p_{1,i})_{1 \leq i \leq \gamma}$ are built as follows:
   - Initially P_0 and P_1 are empty.
   - For each polynomial p such that g_2^p is a sub-term of u_0 at position q, we have that the sub-term of u_1 at position q is also an exponentiation $g_2^{p'}$.
   - If p is not involved in any linear relation with polynomials from the current P_0 and monomials from M, then p is appended to P_0 and p' is appended to P_1. Note that in this case, p' is not involved in any linear relation with polynomials from the current P_1 and monomials from M neither.

Let σ and σ' be the polynomial bijections defined respectively on polynomials p such that g_2^p occurs in term u_0 for σ and on polynomials p such that g_2^p occurs in term u_1 for σ'. These two bijections are defined by:

- For $p_{0,i}$ in P_0 , $p_{0,i}\sigma$ is defined as a fresh monomial $r_{1,i}r_{2,i}r_{3,i}$.
- For $p_{1,i}$ in P_1 , $p_{1,i}\sigma'$ is defined as a fresh monomial $r_{1,i}r_{2,i}r_{3,i}$.
- Let p be a polynomial such that g_2^p occurs in u_0 and such that p is not in P_0. Then by definition of P_0, p is linked via a linear relation to polynomials in P_0 and monomials in M:

  $$p = \sum_{j=1}^{\gamma} \lambda_j p_{0,j} + \sum_{j=1}^{\delta} \mu_j m_j$$

  And we define $p\sigma$ as

  $$p\sigma = \sum_{j=1}^{\gamma} \lambda_j \left(p_{0,j}\sigma\right) + \sum_{j=1}^{\delta} \mu_j m_j$$

- In a similar way, let p be a polynomial such that g_2^p occurs in u_1 and such that p is not in P_1. Then p is linked via a linear relation to polynomials in P_1 and monomials in M:

$$p = \sum_{j=1}^{\gamma} \lambda_j p_{1,j} + \sum_{j=1}^{\delta} \mu_j m_j$$

  And we define $p\sigma'$ as

$$p\sigma' = \sum_{j=1}^{\gamma} \lambda_j \left(p_{1,j}\sigma'\right) + \sum_{j=1}^{\delta} \mu_j m_j$$

Let u be the term $u_0\sigma$. As σ_2 is linear relation preserving, u is equal to $u_1\sigma'$. Then the advantage of $\mathcal{A}$ can be written as:

$$\mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}} = \mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u]\!]_{\mathcal{SE},IG}} + \mathrm{Adv}_{\mathcal{A}}^{[\![u]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}}$$

We now prove that the advantage $\mathrm{Adv}_{\mathcal{A}}^{[\![u]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}}$ is negligible. The proof that $\mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u]\!]_{\mathcal{SE},IG}}$ is also negligible is similar. Let w and w' be the two terms

$$\begin{array}{rcl} w & = & (x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{p_1}, \ldots, g_2^{p_\gamma}) \\ w' & = & (x_1, \ldots, x_\alpha, g_1^{y_1}, \ldots, g_1^{y_\beta}, g_2^{r_{1,1}r_{2,1}r_{3,1}}, \ldots, g_2^{r_{1,\gamma}r_{2,\gamma}r_{3,\gamma}}) \end{array}$$

We build an adversary $\mathcal{B}$ that tries to distinguish $[\![w]\!]_{IG}$ from $[\![w']\!]_{IG}$ and that uses $\mathcal{A}$ as a subroutine. $\mathcal{B}$ works as follows:

1. $\mathcal{B}$ receives as argument a bit-string tuple $(X_1, \ldots, X_\alpha, Y_1, \ldots, Y_\beta, P_1, \ldots, P_\gamma)$ which is either generated by $[\![w]\!]_{IG}$ or by $[\![w']\!]_{IG}$.

2. $\mathcal{B}$ generates bit-string value bs_k for any atomic key k used in u using $\mathcal{KG}$ (these keys are also the ones used in u_0).

3. $\mathcal{B}$ recursively computes a bit-string bs' which is either an evaluation of u (in case $\mathcal{B}$ received as input an evaluation of w') or an evaluation of u_0 (in case $\mathcal{B}$ received as input an evaluation of w). The computation of bs' is done recursively on the structure of u by using the *eval* algorithm:

   - If u is a pair (v, w), then $eval(u) = eval(v) \cdot eval(w)$.
   - If u is an encryption $\{v\}_{key}$, then $eval(u) = \mathcal{E}(eval(v), eval(key))$.
   - If u is an atomic key k, then $eval(u) = bs_k$.
   - If u is an exponent x_i, then $eval(u) = X_i$.
   - If u is an exponentiation $g_1^{y_i}$, then $eval(u) = Y_i$.
   - If u is an exponentiation $g_2^{p_i}$, then $eval(u) = P_i$.

4. Then $\mathcal{B}$ executes $\mathcal{A}$ with bs' as input and returns the same output as $\mathcal{A}$.

We have the following relation among the advantages of $\mathcal{A}$ and $\mathcal{B}$:

$$\mathrm{Adv}_{\mathcal{A}}^{[\![u]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}} = \mathrm{Adv}_{\mathcal{B}}^{[\![w']\!]_{IG},[\![w]\!]_{IG}}$$

As BDDH holds, we apply lemma 5.3 and obtain that the advantage of $\mathcal{B}$ is negligible and hence $\mathrm{Adv}_{\mathcal{A}}^{[\![u]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}}$ is negligible.

Thus $\mathrm{Adv}_{\mathcal{A}}^{[\![u_1]\!]_{\mathcal{SE},IG},[\![u_0]\!]_{\mathcal{SE},IG}}$ is also negligible and we finally obtain that:

$$[\![\mathsf{pat}\,(t_0, K(t_0))]\!]_{\mathcal{SE},IG} \approx [\![\mathsf{pat}\,(t_1, K(t_1))]\!]_{\mathcal{SE},IG}$$

□

It is now easy to obtain our main result by using transitivity of the ≈ relation. Let t_0 and t_1 be two acyclic well-formed terms. Let $\mathcal{SE}$ be an $\mathsf{IND\text{-}CPA}$ secure symmetric encryption scheme and let IG be an instance generator satisfying BDDH. Then we have:

$$[\![t_0]\!]_{\mathcal{SE},IG} \approx [\![\mathsf{pat}\,(t_0, K(t_0))]\!]_{\mathcal{SE},IG} \approx [\![\mathsf{pat}\,(t_1, K(t_1))]\!]_{\mathcal{SE},IG} \approx [\![t_1]\!]_{\mathcal{SE},IG}$$

The previous result states soundness of symbolic equivalence in the computational world. However, the reciprocal (*i.e.*, completeness) is false in general. There are two main problems that prevent completeness. First, the symmetric encryption scheme may allow decryption with the wrong key and output a random bit-string in that case. Then the distributions related to terms $(\{x\}_k, k)$ and $(\{x\}_k, k')$ can be computationally indistinguishable, even though these two terms do not have the same pattern. This can be solved by requiring symmetric encryption to be confusion free [33, 2] or to admit weak key-authenticity tests for expressions [33, 2, 26]. The second problem is that the symmetric encryption scheme can satisfy key concealing (this is ensured by type 0 security in [3]). Then the distributions related to terms $(\{0\}_k, \{0\}_{k'})$ and $(\{0\}_k, \{0\}_k)$ are computationally indistinguishable but these terms are not equivalent even with renaming. To solve this, one can either ask the encryption scheme to be key revealing or modify the pattern definition in order to hide the key name (but the encryption scheme has to be key concealing in order to prove soundness). Soundness and completeness results when symmetric encryption is key and length revealing are given in [5].

The previous proposition considers the case of equivalence and is typically used to verify security of key-exchange protocols. In the next proposition, we are interested in completeness for deducibility. We prove even more than completeness: if t is deducible from E then there exists an efficient algorithm which is able to build an evaluation of t from an evaluation of E with probability 1. This result can be used to verify that a key-agreement protocol can really be implemented in the computational setting: we first check that the shared key is deducible from the knowledge of any participants in the symbolic setting, then applying the following proposition tells us that there exists an efficient algorithm to obtain the shared key from the participant knowledge in the computational setting.

Proposition 5.6 *Let E be a finite set of terms t_1 to t_n and t be a term that does not use any encryption (*e.g.*, a modular exponentiation). If $E \vdash t$ then there exists a polynomial-time (with respect to the security parameter η) algorithm A such that $A\left([\![(t_1, \dots, t_n)]\!]_{\mathcal{SE},IG}\right)$ outputs the evaluation of t using values for exponents and keys that have been generated to compute $[\![(t_1, \dots, t_n)]\!]_{\mathcal{SE},IG}$, i.e.:*

$$(bs, bs') \leftarrow [\![((t_1, \dots, t_n), t)]\!]_{\mathcal{SE},IG} \ : \ A(bs) = bs'$$

Proof. Let t be a term and E be a finite set of terms such that $E \vdash t$. First note that the structure of the proof of $E \vdash t$ does not depend on the security parameter η.

Each deduction rule from the symbolic setting corresponds to an operation which is tractable in the computational setting in polynomial-time in η using a deterministic algorithm (note that the deducibility relation does not give the adversary the ability to encrypt data). Hence it is easy to build the algorithm A by following the structure of a proof of $E \vdash t$. We nevertheless need to restrict ourselves to the case where t does not contain any encryption, as the concrete algorithm for encryption is not deterministic: we indeed have that $\{\{0\}_k\} \vdash \{0\}_k$ while in the computational setting $(bs, bs') \leftarrow [\![(\{0\}_k, \{0\}_k)]\!]_{\mathcal{SE},IG}$ yields two different biststrings bs and bs' as the encryption algorithm is run twice. □

Note that it is not necessary for terms to be well-formed or acyclic in this proposition.

6 Examples of Application

Now we illustrate how proposition 5.2 can be used to prove a key-exchange protocol secure in the computational world.

Our notion of security is strong secrecy of the shared key in the passive setting: the adversary gets to observe messages exchanged between the participants and has to distinguish the shared key from a random group element. In the symbolic world, let us suppose that the exchanged terms were t_1 to t_n and that the shared key is g_2^p. Then security in the symbolic setting holds if:

$$(t_1, ..., t_n, g_2^p) \approx (t_1, ..., t_n, g_2^{r_1 r_2 r_3})$$

where r_1, r_2 and r_3 are three fresh exponent names. It is then possible to apply proposition 5.2 in order to prove security in the computational setting.

We are also interested in executability of key exchange protocols. A protocol is executable if it is feasible for any participant to compute the shared key from his knowledge. Let us again suppose that the exchanged terms are t_1 to t_n and that the shared key is g_2^p. Moreover let $x_i^1, ..., x_i^{k_i}$ be the exponents which are generated by the i^{th} participant. The protocol is executable in the symbolic setting if for any i,

$$t_1, ..., t_n, x_i^1, ..., x_i^{k_i} \vdash g_2^p$$

Executability in the computational world can easily be obtained from here by applying proposition 5.6.

6.1 Joux Protocol

The Joux protocol has been described in section 2. In an execution of this protocol, three messages are sent, corresponding to terms $g_1^{x_1}$, $g_1^{x_2}$ and $g_1^{x_3}$. The shared key is $g_2^{x_1 x_2 x_3}$. Strong secrecy for this key-exchange protocol has been given as an example for our symbolic equivalence notion:

$$(g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, g_2^{x_1 x_2 x_3}) \cong (g_1^{x_1}, g_1^{x_2}, g_1^{x_3}, g_2^{x'_1 x'_2 x'_3})$$

Proposition 5.2 can be applied to show that this protocol is secure in the computational setting if the BDDH assumption holds.

We also verify that this protocol is executable. In the symbolic setting this is the case as we have the following deducibility relation:

$$x_1, g_1^{x_2}, g_1^{x_3} \vdash g_2^{x_1 x_2 x_3}$$

Similar relations hold when permuting the roles of x_1 and x_2 and of x_1 and x_3. Thus proposition 5.6 proves that there exists an efficient algorithm in the computational setting which allows each participant to compute his shared secret key.

6.2 TAK-2 and TAK-3 Protocols

The TAK-2 and TAK-3 protocols are two variants of the Joux protocol which were proposed by Al-Riyami and Paterson in [6]. TAK-1 and TAK-2 are tripartite key-exchange protocols which work in the same way, the only difference lies in the shared key. These protocols uses certificates to provide authentication. However as we are only interested in indistinguishability of the shared key, we use a simplified version of the protocol. Let A, B and C be three participants:

$$\begin{array}{llll} (1)\ A & \rightarrow & B, C & : \ (g_1^{x_1}, g_1^{y_1}) \\ (2)\ B & \rightarrow & A, C & : \ (g_1^{x_2}, g_1^{y_2}) \\ (3)\ C & \rightarrow & A, B & : \ (g_1^{x_3}, g_1^{y_3}) \end{array}$$

In TAK-2, the shared key is $g_2^{x_1 x_2 y_3 + x_1 y_2 x_3 + y_1 x_2 x_3}$. In TAK-3, $g_2^{x_1 y_2 y_3 + y_1 x_2 y_3 + y_1 y_2 x_3}$ is used as shared key. Our simplified version of the two protocols are quite close as we do not make any difference between short-term secrets (y_1, y_2 and y_3) and long-term secrets (x_1, x_2 and x_3). Thus in our setting it is sufficient to analyze one of the protocol, TAK-2 for example.

Security. In the symbolic setting, strong secrecy of the key generated by the TAK-2 protocol comes from the following equivalence (up to renaming). Note that the two equivalent terms are trivially well-formed and acyclic:

$$\begin{gathered} \left(g_1^{x_1}, g_1^{y_1}, g_1^{x_2}, g_1^{y_2}, g_1^{x_3}, g_1^{y_3}, g_2^{x_1 x_2 y_3 + x_1 y_2 x_3 + y_1 x_2 x_3}\right) \\ \cong \\ \left(g_1^{x_1}, g_1^{y_1}, g_1^{x_2}, g_1^{y_2}, g_1^{x_3}, g_1^{y_3}, g_2^{x'_1 x'_2 x'_3}\right) \end{gathered}$$

This equivalence is true because the set of deducible monomials dm is empty for both terms and neither $x_1x_2y_3+x_1y_2x_3+y_1x_2x_3$ in the first term nor $x'_1x'_2x'_3$ in the second term is involved in a linear relation. Hence by using proposition 5.2, we obtain that in the computational setting an adversary that has access to values for $g_1^{x_1}$, $g_1^{y_1}$, $g_1^{x_2}$, $g_1^{y_2}$, $g_1^{x_3}$ and $g_1^{y_3}$ cannot distinguish the shared key $g_2^{x_1x_2y_3+x_1y_2x_3+y_1x_2x_3}$ from a random group element, so the adversary is not able to obtain a single bit of information on the shared key.

Executability. We also verify executability of the protocol. By symmetry we consider the case of A. A generates two exponents x_1 and y_1 and receives two messages corresponding to terms $(g_1^{x_2}, g_1^{y_2})$ and $(g_1^{x_3}, g_1^{y_3})$. Hence executability in the symbolic setting is a consequence of the following deduction:

$$x_1, y_1, g_1^{x_2}, g_1^{y_2}, g_1^{x_3}, g_1^{y_3} \vdash g_2^{x_1x_2y_3+x_1y_2x_3+y_1x_2x_3}$$

Thus proposition 5.6 proves that there exists an efficient algorithm in the computational setting which allows participant A to compute his shared secret key. The same thing holds for B and C.

Active attacks. The TAK protocol family was designed to be secure even in the presence of an active adversary. However, as shown by Shim [37], TAK-2 is vulnerable to active attacks (the other variants are subject to similar attacks). Completely defining a formal model for active adversaries is outside the scope of this paper. Nevertheless the *role* of participant A could be described as follows:

$$\begin{array}{l}\texttt{send}(g_1^{a}, g_1^{\alpha})\\ \texttt{recv}(g_1^{x_B}, g_1^{\beta})\\ \texttt{recv}(g_1^{x_C}, g_1^{\gamma})\end{array}$$

where a is a fresh exponent generated by A, x_B and x_C are variables and $g_1^{\alpha}, g_1^{\beta}, g_1^{\gamma}$ are the public keys which aim at guaranteeing authenticity. The key computed by A corresponds to $K_A = g_2^{ax_B\gamma+a\beta x_C+\alpha x_B x_C}$. An active adversary can substitute x_B and x_C by two fresh names b' and c' yielding an attack: the key computed by A, $g_2^{ab'\gamma+a\beta c'+\alpha b'c'}$, is indeed deducible from the attacker's knowledge $\{g_1^{a}, g_1^{\alpha}, b', g_1^{b'}, g_1^{\beta}, c', g_1^{c'}, g_1^{\gamma}\}$. It follows directly from proposition 5.6 that this symbolic attack can be efficiently implemented by a computational adversary. More generally, active symbolic attacks aiming at *deducing* the key (weak secrecy) correspond to computational attacks. This is not surprising and the converse is obviously not true: it does not follow from our results that a symbolic security proof (in the presence of an active attacker) gives a computational security guarantee.

6.3 A Variant of the Burmester-Desmedt Protocol using Pairings

As an additional example which illustrates the scope of our results, we show how to apply our results to a variant of the group key exchange protocol introduced by Burmester and Desmedt in [14]. The aim of this protocol is to establish a secret key shared among the members of the group. It is scalable as it requires only two rounds and a constant number of modular exponentiation per user. This protocol is only designed for security against passive adversaries.

The Original Burmester-Desmedt Protocol. Consider a network in which members of a group can broadcast messages to each other. Let η be a security parameter and let A_1,A_2,$\cdots$,A_n, for $n \in \mathbb{Z}$, be members of a group. We fix the security parameter η, a finite cyclic group $\mathbb{G}$ of generator g and of prime order q. These parameters $\mathbb{G}$, g and q are published.

- **Round 1:** Each participant A_i samples a random $x_i \in \mathbb{Z}_q$, and broadcasts $Z_i = g^{x_i}$.
- **Round 2:** Each participant A_i broadcasts $X_i = (Z_{i+1}/Z_{i-1})^{x_i} = g^{x_i x_{i+1} - x_{i-1} x_i}$, where the indexes are taken modulo n.
- **Key computation:** Each party A_i computes the shared key $K = g^{\sum_{i=1}^{n} x_i x_{i+1}}$.

The Bilinear Burmester-Desmedt Protocol. Now we define a family of variants of the Burmester-Desmedt protocol. Protocols in this family are parameterized by three integers α, β and γ such that $\alpha + \beta + \gamma = 0$ and either α, β or γ is different from 0. The instance of the protocol corresponding to α, β and γ is denoted by α, β, γ-BBD (Bilinear Burmester-Desmedt).

We still consider a group of n members A_1 to A_n. This time the protocol does not use a single cyclic group but uses a bilinear pairing between two cyclic groups. Hence we fix the security parameter η and two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q with respective generators g_1 and g_2, as well as a pairing operation e from $\mathbb{G}_1 \times \mathbb{G}_1$ to G_2 such that $e(g_1, g_1) = g_2$.

- **Round 1:** Each participant A_i samples a random $x_i \in \mathbb{Z}_q$, and broadcasts $Z_i = g_1^{x_i}$.
- **Round 2:** Each participant A_i broadcasts X_i defined by

$$\begin{array}{rcl} X_i & = & e(Z_{i-2}, Z_{i-1})^{\alpha x_i} e(Z_{i-1}, Z_{i+1})^{\beta x_i} e(Z_{i+1}, Z_{i+2})^{\gamma x_i} \\ & = & g_2^{\alpha x_{i-2} x_{i-1} x_i + \beta x_{i-1} x_i x_{i+1} + \gamma x_i x_{i+1} x_{i+2}} \end{array}$$

  where the indexes are still taken modulo n.
- **Key computation:** Each party A_i computes the shared key

$$K = g_2^{\sum_{i=1}^{n} x_i x_{i+1} x_{i+2}}$$

Security Analysis. We first prove strong secrecy for the shared key in the symbolic setting. This secrecy property is defined as the equivalence between the protocol execution transcript concatenated to the shared key and the transcript concatenated with a random group element from $\mathbb{G}_2$. Hence α, β, γ-BBD verifies strong secrecy of the shared key in the symbolic setting iff the following equivalence holds:

$$(Z_1, ..., Z_n, X_1, ..., X_n, K) \cong (Z_1, ..., Z_n, X_1, ..., X_n, g_2^{r_1 r_2 r_3})$$

In order to obtain this equivalence, we use the following lemma which proves that the exponent used in the key is linearly independent from other exponents if $\alpha + \beta + \gamma = 0$.

Lemma 6.1 *Let α, β, γ and n be four integers. Let V be a real vector space and u_1 to u_n be n linearly independent elements of V. If $\alpha + \beta + \gamma = 0$, then $\sum_{i=1}^{n} u_i$ is linearly independent from the family of vectors $(\alpha u_i + \beta u_{i+1} + \gamma u_{i+2})_i$ (indexes are taken modulo n).*

Proof. Let U be the set of vectors $(\alpha u_i + \beta u_{i+1} + \gamma u_{i+2})_i$ for i between 1 and n. For any vector v in $span(U)$, there exists a unique decomposition $v = \sum_{i=1}^{n} \lambda_i u_i$ and $\sum_{i=1}^{n} \lambda_i$ is equal to 0. Hence $\sum_{i=1}^{n} u_i$ is not in $span(U)$ and is linearly independent from vectors in U. □

A direct consequence of this is strong secrecy of α, β, γ-BBD in the symbolic setting. By applying proposition 5.2, we obtain strong secrecy of the key in the computational setting for a passive adversary.

7 Conclusions and Future Work

We have proposed a first symbolic model to analyze cryptographic protocols which use a bilinear pairing. This model can be used to verify security of well-known key-exchange protocols using pairing like Joux protocol or the TAK-2 and TAK-3 protocol. Moreover our symbolic model consists in an extension of Abadi-Rogaway logic which is computationally sound provided that the encryption scheme and the pairing satisfy classical requirements from provable security. A direct consequence of this soundness result is that the Joux, TAK-2 and TAK-3 protocol are also secure in the computational setting. We also design a variant based on pairings of the Burmester-Desmedt protocol and prove its security against passive adversaries.

This paper only consider passive adversaries. An obvious line for future work is to extend the results to deal with active adversaries. Another interesting follow-up would be to investigate completeness of the extended version of Abadi-Rogaway logic as in [33]. However this would require either to tighten the symbolic model or to use stronger versions of the computational requirements IND-CPA and BDDH.

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 *Lecture Notes in Computer Science*, pages 398–412. Springer, 2006.

[2] M. Abadi and J. Jürjens. Formal eavesdropping and its computational interpretation. In *Proc. 4th International Symposium on Theoretical Aspects of Computer Software (TACS'01)*, volume 2215 of *Lecture Notes in Computer Science*. Springer, 2001.

[3] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. IFIP International Conference on Theoretical Computer Science (IFIP TCS'00)*. Springer, 2000.

[4] 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 *Lecture Notes in Computer Science*, pages 374–396. Springer, 2005.

[5] 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. IEEE, 2005.

[6] S. S. Al-Riyami and K. G. Paterson. Tripartite authenticated key agreement protocols from pairings. In *Proc. 9th IMA International Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 332–359. Springer, 2003.

[7] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *Proc. 10th conference on Computer and Communication Security (CCS'03)*, pages 220–230. ACM, 2003.

[8] G. Bana, P. Mohassel, and T. Stegers. The computational soundness of formal indistinguishability and static equivalence. In *Proc. 11th Asian Computing Science Conference (ASIAN'06)*, volume 4435 of *Lecture Notes in Computer Science*, pages 182–196. Springer, 2008.

[9] P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology – CRYPTO'02*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.

[10] R. Barua, R. Dutta, and P. Sarkar. Extending Joux's protocol to multi party key agreement (extended abstract). In *Proc. 4th International Conference on Cryptology in India (INDOCRYPT'03)*, volume 2904 of *Lecture Notes in Computer Science*, pages 205–217. Springer, 2003.

[11] M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 652–663. Springer, 2005.

[12] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO'01*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[13] 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*, volume 4622 of *Lecture Notes in Computer Science*, pages 482–499. Springer, 2007.

[14] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system (extended abstract). In *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 1994.

[15] R. Canetti and J. Herzog. Universally composable symbolic analysis of mutual authentication and key-exchange protocols. In *Proc. Theory of Cryptography Conference (TCC'06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 380–403. Springer, 2006.

[16] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In *FSTTCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference*, volume 2914 of *Lecture Notes in Computer Science*, pages 124–135. Springer, 2003.

[17] 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/`.

[18] V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *Proc. 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 157–171. Springer, 2005.

[19] A. Datta, A. Derek, J. C. Mitchell, and B. Warinschi. Computationally sound compositional logic for key exchange protocols. In *Proc. 19th IEEE Computer Security Foundations Workshop (CSFW'06)*, pages 321–334. IEEE, 2006.

[20] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 1983.

[21] R. Dutta, R. Barua, and P. Sarkar. Pairing-based cryptographic protocols: A survey. Cryptology ePrint Archive, Report 2004/064, 2004. `http://eprint.iacr.org/`.

[22] F. D. Garcia and P. van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In *Advances in Information and Computer Security. Proc. 1st International Workshop on Security (IWSEC'06)*, volume 4266 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2006.

[23] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proc. 14th Symposium on Theory of Computing (STOC'82)*. ACM, 1982.

[24] P. Gupta and V. Shmatikov. Towards computationally sound symbolic analysis of key exchange protocols. In *Proc. 3rd Workshop on Formal Methods in Security Engineering: From Specifications to Code (FMSE'05)*, pages 23–32. ACM, 2005.

[25] J. Herzog. *Computational soundness for standard assumptions of formal cryptography.* PhD thesis, MIT, 2004.

[26] O. Horvitz and V. D. Gligor. Weak key authenticity and the computational completeness of formal encryption. In *Advances in Cryptology – CRYPTO'03*, volume 2729 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2003.

[27] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. 30th IEEE Symposium on Foundations of Computer Science (FOCS'89)*, pages 248–253, 1989.

[28] R. Janvier, Y. Lakhnech, and L. Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In *Proc. 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 172–185. Springer, 2005.

[29] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Proc. 4th International Symposium on Algorithmic Number Theory (ANTS-IV)*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[30] J. Katz and M. Yung. Scalable protocols for authenticated group key exchange. In *Advances in Cryptology – CRYPTO'03*, volume 2729 of *Lecture Notes in Computer Science*, pages 110–125. Springer, 2003.

[31] 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 *Lecture Notes in Computer Science*, pages 55–66. Springer, 2004.

[32] D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Proc. Theory of Cryptography Conference (TCC'05)*, volume 3378 of *Lecture Notes in Computer Science*, pages 169–187. Springer, 2005.

[33] D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *Journal of Computer Security*, 2004.

[34] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proc. Theory of Cryptography Conference (TCC'04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[35] A. Roy, A. Datta, A. Derek, and J. C. Mitchell. Inductive trace properties for computational security. In *Proc. 7th Workshop on Issues in the Theory of Security (WITS'07)*, 2007.

[36] A. Roy, A. Datta, and J. C. Mitchell. Formal proofs of cryptographic security of Diffie-Hellman based protocols. In *Proceedings of the 3rd Symposium on Trustworthy Global Computing (TGC'07)*, volume 4912 of *Lecture Notes in Computer Science*, pages 312–329. Springer, 2008.

[37] K. Shim. Cryptanalysis of Al-Riyami-Paterson's authenticated three party key agreement protocols. Technical Report 2003/122, Cryptology ePrint Archive, 2003. `http://eprint.iacr.org/`.

Computationally sound implementations of equational theories against passive adversaries☆

Mathieu Baudet[a], Véronique Cortier[b], Steve Kremer[c]

[a] *DCSSI, France*
[b] *Loria/CNRS & INRIA Lorraine projet Cassis, France*
[c] *LSV/ CNRS & INRIA Saclay projet SECSI & ENS Cachan, France*

Abstract

In this paper we study the link between formal and cryptographic models for security protocols in the presence of passive adversaries. In contrast to other works, we do not consider a fixed set of primitives but aim at results for arbitrary equational theories. We define a framework for comparing a cryptographic implementation and its idealization with respect to various security notions. In particular, we concentrate on the computational soundness of static equivalence, a standard tool in cryptographic pi calculi. We present a soundness criterion, which for many theories is not only sufficient but also necessary. Finally, to illustrate our framework, we establish the soundness of static equivalence for the exclusive OR and a theory of ciphers and lists.

1. Introduction

Today's ubiquity of computer networks increases the need for theoretic foundations for cryptographic protocols. For more than twenty years now, two communities separately developed two families of models. Both views have been very useful in increasing the understanding and quality of security protocol design. On the one hand *formal* or *logical* models have been developed, based on the seminal work of Dolev and Yao [2]. These models view cryptographic operations in a rather abstract and idealized way. On the other hand *cryptographic* or *computational* models [3] are closer to implementations: cryptographic operations are modeled as algorithms manipulating bit-strings. Those models cover a large class of attacks, namely all those implementable by a probabilistic polynomial-time Turing machine.

The advantage of formal models is that security proofs are generally simpler and suitable for automatic procedures, even for complex protocols. Unfortunately, the high degree of abstraction and the limited adversary power raise

☆ An extended abstract of this work was published in the proceedings of the ICALP'05 conference [1].

Email addresses: mathieu.baudet@sgdn.gouv.fr (Mathieu Baudet), cortier@loria.fr (Véronique Cortier), kremer@lsv.ens-cachan.fr (Steve Kremer)

questions regarding the security offered by such proofs. Potentially, justifying symbolic proofs with respect to standard computational models has tremendous benefits: protocols can be analyzed using automated tools and still benefit from the security guarantees of the computational model.

For the past few years, a significant research effort has been directed at linking these two approaches. In their seminal work [4], Abadi and Rogaway prove the computational soundness of formal (symmetric) encryption in the case a passive attacker. Since then, many results have been obtained. Each of these results considers a fixed set of primitives, for instance symmetric or public-key encryption. In this paper, we aim at presenting general results for arbitrary equational theories, such as encryption, but also less studied ones, such as groups or exclusive OR. The interest of our approach is not only to develop a general and unified framework for the treatment of cryptographic primitives. Conceptually, it also offers a better understanding of the use of equational theories when modeling the algebraic properties of the primitives. Indeed, for several years, formal models have considered equational theories like the theory of exclusive OR, abelian groups or homomorphic encryption (for a survey on algebraic properties see for instance [5, 6]) in order to model some cryptographic aspects. But it is *a priori* unclear whether "enough" equations have been considered to provide realistic security guarantees. A real attacker might still exploit additional properties of a cryptographic primitive that have not been modeled. Here, we propose a setting and some proof techniques that allow us to formally define and prove that "enough" equations have been considered.

We concentrate on *static equivalence*, a now standard notion originating from the applied pi calculus [7]. Intuitively, static equivalence asks whether an attacker can distinguish between two tuples of messages—later called *frames*—by exhibiting a relation which holds on one tuple but not on the other. Static equivalence provides an elegant means to express security properties on pieces of data, for instance those observed by a passive attacker during the run of a protocol. In the context of active attackers, static equivalence has also been used to characterize process equivalences [7] and off-line guessing attacks [8, 9]. There now exist exact [10] and approximate [11] algorithms to decide static equivalence for a large family of equational theories.

Our first contribution is a general framework for comparing formal and computational models in the presence of a passive attacker. We define the notions of *soundness* and *faithfulness* of a cryptographic implementation with respect to equality, static equivalence and (non-)deducibility. Soundness holds when a formal notion of security has a computational interpretation. For instance, statically equivalent tuples of messages (frames) should be computationally indistinguishable. Conversely, faithfulness holds when every formal attack on a given notion of security can be mapped to an efficient computational attacker. As an illustration, we consider an equational theory modeling Abelian groups with exponents taken over a commutative ring. We show that the soundness of static equivalence implies the hardness of several classical problems in cryptography, notably the decisional Diffie-Hellmann and the RSA problem. Although not completely surprising, this results illustrate well the expressive power of static equivalence defined over tailored equational theories.

Our second contribution is a sufficient criterion for soundness with respect to static equivalence: intuitively the usual computational semantics of terms has to be indistinguishable to an idealized one. We also define and study a useful class of frames, called transparent frames, for arbitrary equational theories. Informally, a frame is transparent if every secret in use is deducible from the frame itself. Transparent frames enjoy notable properties such as a simple characterization of static equivalence and—in the case of uniform distributions—the fact that two statically equivalent transparent frames always yield the same concrete distribution, that is, are indistinguishable in the sense of information theory. This study of transparent frames allows us to exhibit a class of equational theories for which our soundness criterion is necessary.

Our third contribution consists in applying our framework to obtain two first soundness results for static equivalence. The first equational theory that we consider deals with the exclusive OR. This simple but important primitive has been largely used in cryptographic constructions such as the One-Time Pad and in protocols (see [6] for examples). Interestingly, our proof of soundness reflects the unconditional security (in the information-theoretic sense) of the One-Time Pad [12]. Second we consider a theory of symmetric encryption and lists. The result is similar in spirit to the one of Abadi and Rogaway [4]. However, we consider deterministic, length-preserving, symmetric encryption schemes—also known as pseudo-random permutations or ciphers, while Abadi and Rogaway consider probabilistic, symmetric encryption. This choice is motivated by famous examples of ciphers such as DES or AES. In both examples, the specificity of our work is to prove the soundness of a standard formal notion, static equivalence, rather than that of a specialized relation.

Related work.. The study of the link between the formal and the computational approaches for cryptographic protocols started with the seminal work of Abadi and Rogaway [4], in a passive setting. There have been many extensions to the work of Abadi and Rogaway in the passive case, such as studying completeness [13], considering deterministic encryption [14] (a more detailed comparison is provided below), One-Time pad, length-revealing and same-key revealing encryption [12] or allowing composed keys [15] and key-cycles [16].

The first results in an active setting were achieved by Backes, Pfitzmann, and Waidner [17, 18, 19]. These works prove the soundness of a rich language including digital signatures, public-key and symmetric key encryption in the presence of an active attacker for several kind of security properties. Quite similar results were established in more abstract and classical Dolev-Yao models for asymmetric encryption and signatures [20, 21]. While more easily amendable to full automation, these results do not offer universal composability guarantees like the previous ones. However, Canetti and Herzog [22] have recently obtained a similar soundness theorem for a restricted class of protocols—mutual authentication and key exchange protocols using only public-key encryption—which does offer strong composability properties in the universal composability framework. Laud [23] presents an automated procedure for computationally sound proofs of confidentiality in the case of an active attacker and symmetric encryption when the number of sessions is bounded. Datta et al. [24] introduce a symbolic logic that allows cryptographically sound security proof. Recently, Blanchet [25] proposed a computationally sound mechanized prover that relies directly on games transformations, a proof technique commonly used in the cryptographic setting.

Except [25], the previously mentioned results are all dedicated to some fixed set of cryptographic primitives. Here, our goal is not restricted to obtaining some particular soundness result for a given set of primitives and security properties. Rather, we aim at developing a general setting to reason about the adequacy of abstract functional symbols equipped with an equational theory and their corresponding cryptographic implementations. To the best of our knowledge, this approach is new and distinct from existing work. We now discuss some related work concerning the two theories (exclusive OR as well as ciphers and lists) that we have considered to illustrate our framework.

Regarding the soundness of exclusive OR, Backes and Pfitzmann [26] have independently shown an impossibility result in the framework of reactive simulatability, in the presence of an active adversary. They also present a soundness result in the presence of a passive adversary. While we consider the application of exclusive OR only to pure random values, Backes and Pfitzmann deal with arbitrary payloads. It is however not clear how the framework of reactive simulatability in the presence of a passive adversary compares to our framework based on static equivalence.

Concerning the theory of ciphers and list, Laud [14] presents soundness results in the style of Abadi and Rogaway for ciphers. While these results are close to ours, Laud's notion of formal equivalence is apparently more pessimistic than ours regarding the secrecy of encryption keys. For instance, as opposed to [14], we consider that the encryption of a fresh random value by a known key is indistinguishable from a random value—that is, formally, the pair $(\mathsf{enc}(n, k), k)$ is indistinguishable from (n', k). The reason is that, in the absence of tags, each encryption key of a cipher yields a permutation on the space of values. Therefore, if n follows the uniform distribution, such as in our implementation (Section 5.2), so does the term $\mathsf{enc}(n, k)$. Provided a suitable set of equations, static equivalence naturally accounts for this property, whereas there seems to be no natural and immediate way to express the same equivalences using patterns in the style of Abadi and Rogaway. In some sense, the work of Abadi and Warinschi [27] can be seen as an attempt to do so on a fragment of equivalences modeling guessing attacks. Recently, the techniques developed in the present paper have been applied successfully by Abadi, Baudet, and Warinschi [28] to generalize the ideas of [27] and justify a modeling of guessing attacks purely based on static equivalence.

In [14], Laud provides a computationally sound proof system handling both ciphers and exclusive OR in the presence of a passive attacker. This proof system is used to prove the security of several encryption modes including CBC. This approach differs from the one developed here as it aims at direct cryptographic proofs of security (much as in [23, 25]). In comparison, our approach (as in [4, 12, 15, 16, 17, 18, 19, 13, 20, 21]) aims to exhibit a class of protocols for which the absence of formal attacks entails the existence of a computational proof of security.

Further related work.. Since the publication of a preliminary version [1] of this article, several papers have addressed the computational soundness of static equivalence. As already mentioned, Abadi, Baudet, and Warinschi [28] study resistance against offline guessing attacks modelled in terms of static equivalence and use the framework developed in this paper to show the soundness of an equational theory including ciphers, symmetric and asymmetric encryption. In [29], Bana, Mohassel and Stegers argue that the notion of static equivalence is too coarse and not sound for many interesting equational theories. They introduce a general notion of formal indistinguishability relation. This highlights that soundness of static equivalence only holds for a restricted set of well-formed frames (in the same vein Abadi and Rogaway used restrictions to forbid key cycles). They illustrate the unsoundness of static equivalence for modular exponentiation. More recently, Kremer and Mazaré [30] use our framework to define soundness of static equivalence in the presence of an adaptive, rather than purely passive, adversary. They show soundness results of static equivalence for an equational theory modelling modular exponentiation (for a class of well-formed frames, hence not contradicting [29]), as well as symmetric encryption with composed keys which can be computed using modular exponentiation or exclusive or.

The active version of static equivalence is the observational equivalence relation introduced by Milner and Hoare in the early 80s. Intuitively, two processes are equivalent if an observer cannot tell the difference between the two processes. The observer can in particular intercept and send messages to the processes. Comon-Lundh and Cortier [31] have recently shown that observational equivalence between processes in a fragment of the applied pi-calculus [32] implies cryptographic indistinguishability against active attackers, in the context of symmetric encryption. They use an extended version of soundness of static equivalence (called tree soundness) as a key step in their proof.

Outline of the paper.. In the next section, we introduce our abstract and concrete models together with the notions of indistinguishability. We then define the notions of soundness and faithfulness and illustrate some consequences of soundness with respect to static equivalence on groups. In Section 4, we define the ideal semantics of abstract terms, present our soundness criterion, and prove it necessary for a large family of equational theories. As an illustration (Section 5), we prove the soundness for the theories modeling exclusive OR, as well as ciphers and lists. We then conclude in Section 6. An appendix contains detailed proofs of formal lemmas related to static equivalence.

2. Modeling cryptographic primitives with abstract algebras

In this section we introduce some notations and set our abstract and concrete models.

2.1. Abstract algebras

Our abstract models—called *abstract algebras*—consist 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* $\mathcal{S}$, with elements denoted by s, $s_1 \ldots$, and a set $\mathcal{F}$ of *symbols*, written 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 $\mathcal{N}$ of *names*, written a, $b \ldots$, and a set $\mathcal{X}$ of *variables* x, $y \ldots$ We assume that names and variables are given with sorts, and that an infinite number of names and variables are available for each sort. The set of *terms of sort s* is defined inductively by

$$\begin{array}{rlll} T & ::= & & \text{term of sort } s \\ & | & x & \text{variable } x \text{ of sort s} \\ & | & a & \text{name } a \text{ of sort s} \\ & | & f(T_1, \ldots, T_k) & \text{application of symbol } f \in \mathcal{F} \end{array}$$

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 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 $\mathrm{var}(T) = \emptyset$. We may write $\mathrm{var}(T_1, \ldots, T_k)$ instead of $\mathrm{var}(\{T_1, \ldots, T_k\})$ and similarly for names.

A *context* C is a term with holes, or (more formally) a term with distinguished variables. When C is a context with n distinguished variables x_1, $\ldots$, x_n, we may write $C[x_1, \ldots, x_n]$ instead of C in order to show the variables, and when T_1, $\ldots$, T_n are terms we may also write $C[T_1, \ldots, T_n]$ for the result of replacing each variable x_i with the corresponding term T_i.

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* substitutions, that is, substitutions $\sigma = \{x_1 \mapsto T_1, \ldots, x_n \mapsto T_n\}$ for which x_i and T_i have the same sort. 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 var(.) and 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 $\sigma(T) = T\sigma$. If p is a position of T, the expression $T|_p$ denotes the subterm of T at the position p. The expression $T[T']_p$ denotes the term obtained after replacing the subterm in position p of T with T'.

Symbols in $\mathcal{F}$ are intended to model cryptographic primitives, whereas names in $\mathcal{N}$ are used to model secrets, that is, concretely random numbers. The intended behavior of the primitives is described by an equational theory E, that is, an equivalence relation on terms (also written $=_E$) compatible with applications of symbols and well-sorted substitutions:

- for every k-ary symbol f and terms $t_1, \ldots, t_k, t'_1, \ldots, t'_k$ of the appropriate sorts, $\forall i,\ t_i =_E t'_i$ implies that $f(t_1, \ldots, f_k) =_E f(t'_1, \ldots, f'_k)$;

- for every well-sorted substitution σ and terms t, t', if $t =_E t'$ then $t\sigma =_E t'\sigma$.

In the sequel we further require that E is stable under (well-sorted) substitution of names. All the equational theories that we consider in this paper satisfy these properties. For instance, symmetric and deterministic encryption is modeled by the theory E_{enc} generated by the classical equation $E_{\mathsf{enc}} = \{\mathsf{dec}(\mathsf{enc}(x, y), y) = x\}$.

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.

It is often useful to orient equations and work with *rewriting rules* instead of the equational theory. Formally, a *rewriting rule* is an expression $l \to r$ where l and r are two terms of the same sort. Given a set of rewriting rules $\mathcal{R}$ (called *rewriting system*), we write $T \to_{\mathcal{R}} T'$ if there exists a rule $l \to r \in \mathcal{R}$, a position p and a (well-sorted) substitution σ such that $T|_p = l\sigma$ and $T' = T[r\sigma]_p$. We write $\to^*_{\mathcal{R}}$ for the reflexive and transitive closure of $\to_{\mathcal{R}}$, and $=_{\mathcal{R}}$ for its reflexive, symmetric and transitive closure.

Given an equational theory E and a rewriting system $\mathcal{R}$, we write $\to_{\mathcal{R}/E}$ for the relation $=_E\to_{\mathcal{R}}=_E$. We define $\to^*_{\mathcal{R}/E}$ and $=_{\mathcal{R}/E}$ similarly as above. $\mathcal{R}$ is *E-terminating* iff $\to_{\mathcal{R}/E}$ admits no infinite sequence of reductions $T_0 \to_{\mathcal{R}/E} T_1 \to_{\mathcal{R}/E} \ldots T_n \to_{\mathcal{R}/E} \ldots$. It is *$E$-confluent* iff for every $T \to^*_{\mathcal{R}/E} T_1$ and $T \to^*_{\mathcal{R}/E} T_2$, there exist T'_1 and T'_2 such that $T_1 \to^*_{\mathcal{R}/E} T'_1$, $T_2 \to^*_{\mathcal{R}/E} T'_2$, and $T'_1 =_E T'_2$. Finally, $\mathcal{R}$ is *E-convergent* iff it is both E-terminating and E-confluent. When E is the syntactic equality, this yields the usual notions of termination, confluence and convergence.

2.2. Frames, deducibility and static equivalence

We use frames [7, 10] to represent sequences of messages observed by an attacker, for instance during the execution of a protocol. Formally, a (closed) *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 (closed) frames $\varphi = \nu\tilde{a}.\{x_1 = T_1, \ldots, x_n = T_n\}$ which restrict every name in use, that is, for which $\tilde{a} = \mathrm{names}(T_1, \ldots, T_n)$. 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* $\sigma = \{x_1 \mapsto T_1, \ldots, x_n \mapsto T_n\}$.

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 $\varphi_1 = \nu k_1, k_2, k_3, k_4.\,\{x_1 = \mathsf{enc}(k_1, k_2),\ x_2 = \mathsf{enc}(k_4, k_3),\ x_3 = k_3\}$: the name k_4 is deducible from φ_1 since $\mathsf{dec}(x_2, x_3)\varphi_1 =_{E_{\mathsf{enc}}} k_4$ but neither k_1 nor k_2 are deducible.

Deducibility is not always sufficient to account for the knowledge of an attacker. For instance, it lacks partial information on secrets. Indeed, if we consider a naive vote protocol where agents simply send their vote (0 or 1) encrypted under some key, the security problem is not whether an attacker can learn the values of 0 or 1, but rather whether an attacker can tell the difference between a message that contains the vote 0 and a message that contains the vote 1. That is why another classical notion in formal methods is *static equivalence*.

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 $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi_1, \varphi_2) = \emptyset$, $M\varphi_1 =_E N\varphi_1$ if and only if $M\varphi_2 =_E N\varphi_2$.

For instance, the two frames $\nu k.\, \{x = \mathsf{enc}(0, k)\}$ and $\nu k.\, \{x = \mathsf{enc}(1, k)\}$ are statically equivalent with respect to E_{enc}. However the two frames

$$\nu k.\{x = \mathsf{enc}(0, k),\ y = k\} \text{ and } \nu k.\{x = \mathsf{enc}(1, k),\ y = k\}$$

are not (consider the test $\mathsf{dec}(x, y) \overset{?}{=} 0$), although the set of terms that can be deduced from both frames is the same (0 and 1 are two constants known by the attacker).

2.3. 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 bit-strings $[\![s]\!]_A \subseteq \{0, 1\}^*$ for each sort $s \in \mathcal{S}$;
- an effective procedure implementing a function $[\![f]\!]_A : [\![s_1]\!]_A \times \ldots \times [\![s_k]\!]_A \to [\![s]\!]_A$ for each symbol $f \in \mathcal{F}$ with $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$;
- an effective procedure for deciding a 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 bit-strings); 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 $x \xleftarrow{R} [\![s]\!]_A$; the drawing may not follow a uniform distribution, but no $=_{A,s}$-equivalence class should have probability 0.

Assume a fixed $(\mathcal{S}, \mathcal{F})$-computational algebra A. We associate to each (closed) frame $\varphi = \{x_1 = T_1, \ldots, x_n = T_n\}$ 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: $\widehat{f(T'_1, \ldots, T'_m)} = [\![f]\!]_A(\widehat{T'_1}, \ldots, \widehat{T'_m})$; using the values $\widehat{a}$ defined at step 1 for names.
3. return the value $\widehat{\psi} = \{x_1 = \widehat{T_1}, \ldots, x_n = \widehat{T_n}\}$.

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 natural way: $e_1, \ldots, e_n \xleftarrow{R} [\![T_1, \ldots, T_n]\!]_A$ denotes the drawing

$$\{x_1 = e_1, \ldots, x_n = e_n\} \xleftarrow{R} [\![\{x_1 = T_1, \ldots, x_n = T_n\}]\!]_A$$

for appropriate variables $x_1, \ldots, x_n$. We also generalize the notation to (tuples of) terms with variables, by specifying a concrete value for each of them: $[\![\cdot]\!]_{A,\{x_1=e_1,\ldots,x_n=e_n\}}$. Notice that when a term or a frame contains no names, the translation is deterministic; in this case, we use the same notation to denote the distribution and its unique value.

In the rest of the paper we focus on asymptotic notions of cryptographic security and consider families of computational algebra (A_η) indexed by a complexity parameter $\eta \geq 0$. (This parameter η might be thought as the size of keys and other secret values.) 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 the algebraic operations (i.e. the functions associated to symbols, the congruence relation $=_{A,s}$, and the drawing functions) are computable by uniform, probabilistic polynomial-time algorithms in the complexity parameter η. This ensures that the concrete semantics of every (fixed) term or frame is efficiently computable (in the same sense).

Families of distributions (*ensembles*) over concrete frames benefit from the usual notion of cryptographic indistinguishability. Intuitively, two families of distributions (ψ_η) and (ψ'_η) are *indistinguishable*, written $(\psi_\eta) \approx (\psi'_\eta)$, iff no probabilistic polynomial-time adversary $\mathcal{A}$ can guess whether he is given a sample from ψ_η or ψ'_η with a probability significantly greater than $\frac{1}{2}$. Formally, we ask the *advantage* of $\mathcal{A}$,

$$\mathrm{Adv}^{\mathrm{IND}}(\mathcal{A}, \eta, \psi_\eta, \psi'_\eta) = \mathbb{P}[\widehat{\psi} \xleftarrow{R} \psi_\eta \;:\; \mathcal{A}(\eta, \widehat{\psi}) = 1] - \mathbb{P}[\widehat{\psi} \xleftarrow{R} \psi'_\eta \;:\; \mathcal{A}(\eta, \widehat{\psi}) = 1]$$

to be a *negligible* function of η. We recall that a function f is said *negligible* if for any integer $n > 0$, there exists η_0 such that $f(\eta) \leq \eta^{-n}$ for any $\eta \geq \eta_0$. (Note that we regard negative functions as negligible here.)

A function $f(\eta)$ is *overwhelming* iff $1 - f(\eta)$ is negligible. A family of distributions (ψ_η) is *collision-free* (with respect to the family of congruences $=_{A_\eta}$) iff the probability of collision between two random elements from ψ_η, that is, $\mathbb{P}[\, e_1, e_2 \xleftarrow{R} \psi_\eta \;:\; e_1 =_{A_\eta} e_2]$, is a negligible function of η. Note that, by classical properties of probability, this is equivalent to requiring that the probability of sampling any given e_0 from ψ_η (modulo $=_{A_\eta}$) is negligible, that is, the function $\sup_{e_0} \mathbb{P}\left[e \xleftarrow{R} \psi_\eta \;:\; e =_{A_\eta} e_0\right]$ is bounded by a negligible function of η.

By convention, the adversaries considered in this paper are given access implicitly to the complexity parameter η and to as many fresh random coins as needed.

3. Comparing 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.

3.1. 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 $\mathbb{P}[\, e_1, e_2 \overset{R}{\longleftarrow} [\![T_1,\, T_2]\!]_{A_\eta} \ :\ e_1 =_{A_\eta} e_2]$ 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 $\mathbb{P}[\, e_1, e_2 \overset{R}{\longleftarrow} [\![T_1,\, T_2]\!]_{A_\eta} \ :\ e_1 =_{A_\eta} e_2]$ 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 $\mathcal{A}$, $\mathbb{P}[\phi, e \overset{R}{\longleftarrow} [\![\varphi, T]\!]_{A_\eta} : \mathcal{A}(\phi) =_{A_\eta} e]$ 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 $\mathbb{P}[\phi, e \overset{R}{\longleftarrow} [\![\varphi, T]\!]_{A_\eta} \ :\ \mathcal{A}(\phi) =_{A_\eta} e]$ is overwhelming.

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 $\mathbb{P}[\, e_1, e_2 \overset{R}{\longleftarrow} [\![T_1,\, T_2]\!]_{A_\eta} \ :\ e_1 =_{A_\eta} e_2] = 1$;
- *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.

The fact that the first two unconditional notions are stronger than their computational counterparts is clear from the definitions. As for the unconditional $\not\vdash_E$-soundness, observe that if the drawings for φ and T are independent, and the drawing $(\xleftarrow{R} [\![T]\!]_{A_\eta})$ is collision-free, then any adversary $\mathcal{A}$ has negligible probability of retrieving the value of T:

$$\begin{aligned}
&\mathbb{P}[\phi, e \xleftarrow{R} [\![\varphi, T]\!]_{A_\eta} \,:\, \mathcal{A}(\phi) =_{A_\eta} e] \\
&\qquad = \mathbb{P}[\phi \xleftarrow{R} [\![\varphi]\!]_{A_\eta}, e \xleftarrow{R} [\![T]\!]_{A_\eta} \,:\, \mathcal{A}(\phi) =_{A_\eta} e] \\
&\qquad\qquad \leq \sup_{e_0} \mathbb{P}[e \xleftarrow{R} [\![T]\!]_{A_\eta} \,:\, e =_{A_\eta} e_0]
\end{aligned}$$

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: for instance Abadi and Rogaway [4] do forbid encryption cycles (see Section 5.2).

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.*

Proof.

1. Suppose $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi)$ and $\varphi \vdash_E T$, that is, there exists M such that $\mathrm{var}(M) \subseteq \mathrm{dom}(\varphi)$, $\mathrm{names}(M) \cap \mathrm{names}(\varphi) = \emptyset$, and $M\varphi =_E T$. We define an adversary $\mathcal{A}$ which can deduce $[\![T]\!]$ from $[\![\varphi]\!]$ as follows: given the concrete frame $\phi = \{x_i = e_i\}$, $\mathcal{A}$ returns a sample $e \xleftarrow{R} [\![M]\!]_{A_\eta,\phi}$. As $(A_\eta)_{\eta \geq 0}$ is $=_E$-sound and $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi)$, $\mathcal{A}$'s probability of success is greater than 1 minus a negligible function.

2. Suppose $\varphi_1 \not\approx_E \varphi_2$: there exist two terms M and N such that $\mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi_1)$, $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi_1, \varphi_2) = \emptyset$, and for instance $M\varphi_1 =_E N\varphi_1$ whereas $M\varphi_2 \neq_E N\varphi_2$. Let $\mathcal{A}$ be the adversary that tests, given η and ϕ, whether $[\![M]\!]_{A_\eta,\phi} =_{A_\eta} [\![N]\!]_{A_\eta,\phi}$, and returns the result of the test. $\mathcal{A}$ runs in polynomial-time and by $=_E$-soundness and $=_E$-faithfulness, its advantage is 1 minus a negligible function. □

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.

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 $\not\vdash_E$-sound;*
3. *Assume the implementations for the symbols h_s are* collision-resistant, *that is, assume 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, $\not\vdash_E$-faithful and $\approx_E$-faithful.

Proof.

1. Let T_1, T_2 be two terms of sort s such that $T_1 \neq_E T_2$. Consider the frame $\varphi = \{x_1 = \mathsf{h}_s(T_1,k),\ x_2 = \mathsf{h}_s(T_2,k)\}$ where k is a fresh name of sort *Key*. As $T_1 \neq_E T_2$ and h_s is free, we have $\varphi \approx_E \{x_1 = n, x_2 = n'\}$ where n, n' are two distinct fresh names of sort *Hash* (Proposition 17 of Appendix A). By assumption, this entails $[\![\varphi]\!] \approx [\![\{x_1 = n,\, x_2 = n'\}]\!]$. In particular, since $(\xleftarrow{R} [\![Hash]\!]_{A_\eta})$ is collision-free, the quantity

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

   is negligible.

2. Let φ be a frame and T a closed term of sort s such that names$(T) \subseteq$ names(φ) and $\varphi \not\vdash_E T$. We let $\varphi_0 = \varphi \cup \{x = \mathsf{h}_s(T,k), y = k\}$ and $\varphi_1 = \varphi \cup \{x = n, y = k\}$ where x, y are fresh variables, k is a fresh name of sort *Key*, n is a fresh name of sort *Hash*. As $\varphi \not\vdash_E T$, we have $\varphi_0 \approx_E \varphi_1$ (Proposition 18 of Appendix A). Thus by assumption, $[\![\varphi_0]\!] \approx [\![\varphi_1]\!]$.

   By contradiction, suppose that there exists a polynomial-time adversary $\mathcal{A}$ able to deduce $[\![T]\!]$ from $[\![\varphi]\!]$ concretely with non-negligible probability of success. We build an adversary $\mathcal{B}$ that distinguishes between $[\![\varphi_0]\!]$ and $[\![\varphi_1]\!]$ as follows: let ϕ be the sample from $[\![\varphi_b]\!]_\eta$ to be analyzed, where $b \in \{0,1\}$. Let $\widehat{T}$ be the answer of $\mathcal{A}$ when given the restriction of ϕ

to $\mathrm{dom}(\varphi)$. $\mathcal{B}$ returns 0 if $x\phi =_{A_\eta} [\![\mathsf{h}_s]\!]_{A_\eta}(\widehat{T}, y\phi)$, and 1 otherwise. By definition, the advantage of $\mathcal{B}$ is

$$\begin{aligned}
&\mathbb{P}[\phi \xleftarrow{R} [\![\varphi_0]\!]_\eta \ : \ \mathcal{B}(\eta, \phi) = 0] - \mathbb{P}[\phi \xleftarrow{R} [\![\varphi_1]\!]_\eta \ : \ \mathcal{B}(\eta, \phi) = 0] \\
&\quad = \quad \mathbb{P}[\phi \xleftarrow{R} [\![\varphi_0]\!]_\eta; \widehat{T} \xleftarrow{R} \mathcal{A}(\phi|_{\mathrm{dom}(\varphi)}) \ : \ x\phi =_{A_\eta} [\![\mathsf{h}_s]\!]_{A_\eta}(\widehat{T}, y\phi)] \\
&\qquad - \mathbb{P}[\phi \xleftarrow{R} [\![\varphi_1]\!]_\eta; \widehat{T} \xleftarrow{R} \mathcal{A}(\phi|_{\mathrm{dom}(\varphi)}) \ : \ x\phi =_{A_\eta} [\![\mathsf{h}_s]\!]_{A_\eta}(\widehat{T}, y\phi)] \\
&\quad \geq \quad \mathbb{P}[\phi, e \xleftarrow{R} [\![\varphi_0, T]\!]_\eta; \widehat{T} \xleftarrow{R} \mathcal{A}(\phi|_{\mathrm{dom}(\varphi)}) \ : \ \widehat{T} = e)] \\
&\qquad - \mathbb{P}[\phi \xleftarrow{R} [\![\varphi_1]\!]_\eta; \widehat{T} \xleftarrow{R} \mathcal{A}(\phi|_{\mathrm{dom}(\varphi)}) \ : \ x\phi =_{A_\eta} [\![\mathsf{h}_s]\!]_{A_\eta}(\widehat{T}, y\phi)]
\end{aligned}$$

In the last probability expression, observe that $x\phi$ is drawn from the distribution $(\xleftarrow{R} [\![\mathit{Hash}]\!]_{A_\eta})$ independently from $\widehat{T}$ and $y\phi$. Hence, as the distribution $(\xleftarrow{R} [\![\mathit{Hash}]\!]_{A_\eta})$ is collision-free, the advantage of $\mathcal{B}$ is non-negligible; contradiction.

3. Let T_1 and T_2 be two terms of sort s such that $T_1 =_E T_2$. Consider the same frame as before: $\varphi = \{x_1 = \mathsf{h}_s(T_1, k),\ x_2 = \mathsf{h}_s(T_2, k)\}$. As $T_1 =_E T_2$ and h_s is free, we have $\varphi \approx_E \{x_1 = n,\ x_2 = n\}$ where n is a fresh name of sort *Hash* (Proposition 19 of Appendix A). By assumption this entails that $[\![\varphi]\!] \approx [\![\{x_1 = n,\ x_2 = n\}]\!]$ thus

$$\mathbb{P}\left[e_1', e_2' \xleftarrow{R} [\![\mathsf{h}_s(T_1, k),\ \mathsf{h}_s(T_2, k)]\!]_{A_\eta} \ : \ e_1' =_{A_\eta} e_2'\right] \geq 1 - \epsilon_\eta$$

where ϵ_η is a negligible function. As the implementation of h_s is collision-resistant, we deduce that

$$\mathbb{P}\left[e_1, e_2 \xleftarrow{R} [\![T_1,\ T_2]\!]_{A_\eta} \ : \ e_1 \neq_{A_\eta} e_2\right]$$

is negligible. Other properties follow from Proposition 1. □

3.2. $\approx_E$-soundness implies classical assumptions on groups

In this section we present some interesting consequences of $\approx_E$-soundness. Inspired by the work of Hohenberger and Rivest on pseudo-freeness [33, 34], we prove that several standard cryptographic assumptions on groups are implied by the soundness of static equivalence. We concentrate on abelian groups as these are more relevant for cryptographic applications. We believe that similar techniques would apply for non-commutative groups as well.

We model an abelian group G with exponents taken over a commutative ring A by an abstract algebra over the following signature:

$$\begin{array}{ccl@{\qquad}ccl}
* & : & G \times G \to G & - & : & A \to A \\
1_G & : & G & \cdot & : & A \times A \to A \\
+ & : & A \times A \to A & 1_A & : & A \\
0 & : & A & \mathsf{exp} & : & G \times A \to G
\end{array}$$

We use the infix notation for the operators $*$, $\cdot$, $+$, and write g^a to denote $\mathsf{exp}(g, a)$. Note that the inverse operation on G is represented here by $g \mapsto \mathsf{exp}(g, -(1_A)) = g^{-(1_A)}$. We consider the equational theory E_{G} generated by the following equations (where x, y, z are variables of sort G, and u, v, w variables of sort A):

$$\begin{array}{rclcrcl}
u+v &=& v+u & \qquad & x*y &=& y*x \\
u+(v+w) &=& (u+v)+w & & x*(y*z) &=& (x*y)*z \\
u+0_A &=& u & & x*1_G &=& x \\
u+(-u) &=& 0_A & & (x^u)^v &=& x^{(u\cdot v)} \\
u\cdot v &=& v\cdot u & & x^u*x^v &=& x^{u+v} \\
u\cdot(v\cdot w) &=& (u\cdot v)\cdot w & & x^{1_A} &=& x \\
u\cdot 1_A &=& u & & x^{0_A} &=& 1_G \\
(u+v)\cdot w &=& u\cdot w+v\cdot w & & (x*y)^u &=& x^u*y^u
\end{array}$$

We now recall several classical problems on groups. For cryptographic applications, it is desirable that these problems be *hard*, that is, not feasible by any probabilistic polynomial-time adversary:

- *discrete logarithm* (DL) problem: given g and g', find a, such that $g^a = g'$;
- *computational Diffie-Hellman* (CDH) problem: given g, g^a and g^b, find g^{ab};
- *decisional Diffie-Hellman* (DDH) problem: given g, g^a and g^b, distinguish g^{ab} from a random element g^c;
- *RSA* problem: given elements a and g^a, find g.

A more detailed presentation of these hard problems can be found in [35].

Assume a family of computational algebras (A_η) over the signature above such that (A_η) is $\approx_{E_{\mathsf{G}}}$-sound, at least for some subset of well-formed frames WF. Consider the two frames

$$\varphi_1 = \nu g, a, b.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^{a\cdot b}\} \text{ and}$$
$$\varphi_2 = \nu g, a, b, c.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^c\}.$$

and assume that φ_1, $\varphi_2 \in WF$. Then no probabilistic polynomial-time adversary $\mathcal{A}$ can solve the DDH problem in (A_η) with non-negligible probability.

Indeed, as suggested in [7], the question of (computationally) distinguishing these two frames exactly encodes the DDH problem. Given the equational theory E_{G}, we prove the formal equivalence $\varphi_1 \approx_{E_{\mathsf{G}}} \varphi_2$ (Lemma 21 of Appendix B). Thus, by $\approx_{E_{\mathsf{G}}}$-soundness, the DDH problem is hard in (A_η).

Clearly, if one can solve the DL problem, one can also solve the CDH problem, which itself allows us to solve the DDH problem. Therefore, the hardness of DDH implies the hardness of the two other problems.

In a similar way, we see that $\approx_{E_{\mathsf{G}}}$-soundness on an augmented signature implies the hardness of RSA. Instead of directly encoding the RSA problem, we introduce a slightly weaker decision problem, whose hardness implies the hardness of RSA. The encoding of this problem requires the extension of the signature by a unary function symbol $\mathsf{h} : G \to \mathit{Hash}$, adding no equation to the theory. Consider the two frames

$$\varphi_3 = \nu g, a.\{x_1 = g^a, x_2 = a, x_3 = \mathsf{h}(g)\} \text{ and}$$
$$\varphi_4 = \nu g, a, h.\{x_1 = g^a, x_2 = a, x_3 = h\}.$$

We prove that $\varphi_3 \approx_{E_\mathsf{G}} \varphi_4$ in Lemma 22 of Appendix B. As above, if an implementation (A_η) is $\approx_{E_\mathsf{G}}$-sound of for some subset of well-formed frames WF including φ_3 and φ_4, then the RSA problem cannot be efficiently solved in (A_η). Indeed, any adversary $\mathcal{A}$ to the RSA-problem can be turned to an (equally efficient) adversary against $(\llbracket\varphi_3\rrbracket_{A_\eta}) \approx (\llbracket\varphi_4\rrbracket_{A_\eta})$ simply as follows: given a sample $\{x_1 = e_1, x_2 = e_2, x_3 = e_3\}$ from either side, let e be the result of $\mathcal{A}$ applied on η, e_1 and e_2; return 1 ("left-hand side") if $\llbracket h\rrbracket_{A_\eta}(e)$ equals to e_3, 0 otherwise.

An interesting open question is whether $\approx_{E_\mathsf{G}}$-soundness implies or is implied by Rivest's notion of pseudo-free groups [34], or equivalently [36], the strong RSA property. We conjecture that the two notions are in fact incomparable. Indeed, on the one hand, our notion implies the hardness of DDH, which remains an open question for strong RSA. On the other hand, pseudo-freeness and strong RSA deal with a form of adaptive attackers while our model is purely non-adaptive.

4. A sufficient (and often necessary) criterion for $\approx_E$-soundness

We now present useful results for proving $\approx_E$-soundness properties in general. Notably, we provide a sufficient criterion for $\approx_E$-soundness in Section 4.1 and prove it necessary under additional assumptions in Section 4.2.

4.1. Ideal semantics and $\approx_E$-soundness criterion

Given an implementation of the primitives, we have defined in Section 2.3 the concrete semantics $\llbracket\varphi\rrbracket_{A_\eta}$ associated to every frame φ . We now define the *ideal semantics* of a frame φ, intuitively as the conditional distribution over all the concrete values (in the appropriate space) that pass every formal test satisfied by φ.

Specifically, for every frame φ, we define the *tests* of φ to be

$$\mathrm{test}(\varphi) = \{(M, N) \mid \mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi),\ \mathrm{names}(M, N) \cap \mathrm{names}(\varphi) = \emptyset\}.$$

We let $\mathrm{eq}_E(\varphi)$ be the set of tests that are true in φ:

$$\mathrm{eq}_E(\varphi) = \{(M, N) \in \mathrm{test}(\varphi) \mid M\varphi =_E N\varphi\}$$

Note that, by definition, $\varphi \approx_E \varphi'$ iff $\mathrm{eq}_E(\varphi) \cap \mathrm{test}(\varphi') = \mathrm{eq}_E(\varphi') \cap \mathrm{test}(\varphi)$.

Let (A_η) be a family of computational algebras, $\varphi = \{x_1 = T_1, \ldots, x_n = T_n\}$ be a frame, and s_i be the sort of x_i. We define the set of eligible, well-formed values for φ by

$$\mathrm{Val}_{A_\eta}(\varphi) = \big\{\{x_1 = e_1, \ldots, x_n = e_n\} \mid (e_1, \ldots, e_n) \in \llbracket s_1\rrbracket_{A_\eta} \times \cdots \times \llbracket s_n\rrbracket_{A_\eta}\big\}$$

and write $\phi \xleftarrow{R} [\![\varphi]\!]^{\mathrm{val}}_{A_\eta}$ for the process of drawing a random value $\phi = \{x_1 = e_1, \ldots, x_n = e_n\}$ from $\mathrm{Val}_{A_\eta}(\varphi)$ using the drawings $e_i \xleftarrow{R} [\![s_i]\!]_{A_\eta}$ in the natural way.

Consider the following subset of concrete frames, intuitively, that pass all the valid tests of φ:

$$\begin{aligned}\mathrm{Val}'_{A_\eta}(\varphi) = \big\{\phi \in \mathrm{Val}_{A_\eta}(\varphi) \mid \forall (M,N) \in \mathrm{eq}_E(\varphi),\\ \mathbb{P}\Big[u, v \xleftarrow{R} [\![M,N]\!]_{A_\eta,\{x_1=e_1,\ldots,x_n=e_n\}} \ : \ u = v\Big] = 1\big\}\end{aligned}$$

Note that, provided that (A_η) is unconditionally $=_E$-sound, $\mathrm{Val}'_{A_\eta}(\varphi)$ is non-empty as it contains at least the values given by the usual semantics of φ.

Definition 3 (Ideal semantics). Let (A_η) be an unconditionally $=_E$-sound family of computational algebras and φ be a frame. The *ideal semantics* of φ is the family of the distributions $[\![\varphi]\!]^{\mathrm{ideal}}_{A_\eta}$ obtained by conditionning each distribution $[\![\varphi]\!]^{\mathrm{val}}_{A_\eta}$ to the set of values $\mathrm{Val}'_{A_\eta}(\varphi)$. In other words, the probability to draw $\phi \in \mathrm{Val}_{A_\eta}(\varphi)$ is

$$\mathbb{P}[\phi \leftarrow [\![\varphi]\!]^{\mathrm{ideal}}_{A_\eta}] = \begin{cases} 0 & \text{if } \phi \notin \mathrm{Val}'_{A_\eta}(\varphi) \\ \frac{1}{V}\, \mathbb{P}[\phi \xleftarrow{R} [\![\varphi]\!]^{\mathrm{val}}_{A_\eta}] & \text{otherwise} \end{cases}$$

where $V = \mathbb{P}[\phi_0 \xleftarrow{R} [\![\varphi]\!]^{\mathrm{val}}_{A_\eta} \ : \ \phi_0 \in \mathrm{Val}'(\varphi)]$.

We say that (A_η) *has uniform distributions* if and only if for every η and every sort s, $[\![s]\!]_{A_\eta}$ is a finite set, $=_{A_\eta,s}$ is the usual equality, and the distribution associated to s by A_η is the uniform one over $[\![s]\!]_{A_\eta}$.

By classical property of conditional probabilities, we note that in the case of uniform distributions, the ideal semantics of a frame φ coincides with the family of uniform distributions over the (finite, non-empty) sets $\mathrm{Val}'_{A_\eta}(\varphi)$.

For instance, let $\varphi = \nu n_1, n_2.\{x_1 = n_1, x_2 = n_2\}$ with n_1 and n_2 of sort s. Then, given that E is stable by substitution of names, we have that $\mathrm{eq}_E(\varphi) = \{(M,N) \in \mathrm{test}(\varphi) \mid M =_E N\}$. By unconditional $=_E$-soundness, we deduce that $[\![\varphi]\!]^{\mathrm{ideal}}_{A_\eta}$ is simply the uniform distribution over $[\![s]\!]_{A_\eta} \times [\![s]\!]_{A_\eta}$.

We now state our $\approx_E$-soundness criterion: intuitively, the two semantics, concrete and ideal, should be indistinguishable.

Proposition 3 ($\approx_E$-soundness criterion). *Let (A_η) be an unconditionally $=_E$-sound family of computational algebras. Assume that for every frame φ it holds that $([\![\varphi]\!]_{A_\eta}) \approx ([\![\varphi]\!]^{\mathrm{ideal}}_{A_\eta})$. Then (A_η) is $\approx_E$-sound.*

Proof. Let $\varphi_1 \approx_E \varphi_2$. The equality $\mathrm{eq}_E(\varphi_1) \cap \mathrm{test}(\varphi_2) = \mathrm{eq}_E(\varphi_2) \cap \mathrm{test}(\varphi_1)$ entails $\mathrm{Val}'_{A_\eta}(\varphi_1) = \mathrm{Val}'_{A_\eta}(\varphi_2)$, thus the distributions $[\![\varphi_1]\!]^{\mathrm{ideal}}_{A_\eta}$ and $[\![\varphi_2]\!]^{\mathrm{ideal}}_{A_\eta}$ are equal. We use the transitivity of the indistinguishability relation $\approx$ to conclude: $([\![\varphi_1]\!]_{A_\eta}) \approx ([\![\varphi_1]\!]^{\mathrm{ideal}}_{A_\eta}) = ([\![\varphi_2]\!]^{\mathrm{ideal}}_{A_\eta}) \approx ([\![\varphi_2]\!]_{A_\eta})$. □

4.2. Transparent frames

In this section we show that our soundness criterion is necessary for a general class of equational theories, called *transparent* theories. In those theories, each frame can be associated to an equivalent *transparent frame* (defined below), which is easier to analyze.

Definition 4 (Transparent frames). A frame φ is *transparent* for an equational theory E if each of its subterms is deducible from φ in E.

Example 1. In the theory E_{enc}, the frame $\varphi_1 = \{x_1 = \mathsf{enc}(\mathsf{enc}(k_4, k_3), k_1),\ x_2 = \mathsf{enc}(k_1, k_2),\ x_3 = k_2\}$ is not transparent, as neither k_3 nor k_4 are deducible, but the frame $\overline{\varphi_1} = \{x_1 = \mathsf{enc}(n_1, k_1), x_2 = \mathsf{enc}(k_1, k_2), x_3 = k_2\}$ is.

The following proposition finitely characterizes the equations verified by a transparent frame.

Proposition 4. *Let φ be a transparent frame for E. Then, φ is of the form*

$$\varphi = \{x_1 = C_1[a_1, \ldots, a_m], \ldots, x_n = C_n[a_1, \ldots, a_m]\}$$

where $C_1, \ldots, C_n$ are (not necessarily linear) contexts such that $\mathrm{names}(C_1, \ldots, C_n) = \emptyset$, $C_1[a_1, \ldots, a_m], \ldots, C_n[a_1, \ldots, a_m]$ are closed and, $a_1, \ldots, a_m$ are distinct deducible names: $\varphi \vdash_E a_i$.

For each a_i, let M_{a_i} be a term such that $\mathrm{var}(M_{a_i}) \subseteq \{x_1, \ldots, x_n\}$, $\mathrm{names}(M_{a_i}) \cap \mathrm{names}(\varphi) = \emptyset$ and $M_{a_i}\varphi =_E a_i$. Then every equation which holds in φ is a logical consequence of E and the equations $x_j = C_j[M_{a_1}, \ldots, M_{a_m}]$, written

$$E \cup \{x_j = C_j[M_{a_1}, \ldots, M_{a_m}] \mid 1 \leq j \leq n\} \models \mathrm{eq}_E(\varphi).$$

By logical consequence, we refer to the usual first-order theory of equality, where the variables x_1, …, x_n are considered here as constants.

Proof. Let $(M, N) \in \mathrm{eq}_E(\varphi)$. By definition, we have $M\varphi =_E N\varphi$, that is, $M\{x_j \mapsto C_j[a_1, \ldots, a_m]\}_{1 \leq j \leq n} =_E N\{x_j \mapsto C_j[a_1, \ldots, a_m]\}_{1 \leq j \leq n}$. Since E is stable by substitution of names, we obtain

$$M\{x_j \mapsto C_j[M_{a_1}, \ldots, M_{a_m}]\}_{1 \leq j \leq n} =_E N\{x_j \mapsto C_j[M_{a_1}, \ldots, M_{a_m}]\}_{1 \leq j \leq n}.$$

Using the equalities $x_j = C_j[M_{a_1}, \ldots, M_{a_m}]$ and by transitivity, we obtain $\{x_j = C_j[M_{a_1}, \ldots, M_{a_m}] \mid 1 \leq j \leq n\} \cup E \models M = N$. □

Another nice and useful property of transparent frames is that their concrete and ideal semantics coincide.

Proposition 5. *Let (A_η) be an unconditionally $=_E$-sound family of computational algebras, having uniform distributions. Let φ be a transparent frame. The concrete and the ideal semantics of φ yield the same family of distributions: for all η, $[\![\varphi]\!]_{A_\eta} = [\![\varphi]\!]^{\mathrm{ideal}}_{A_\eta}$.*

Proof. Let $\varphi = \{x_1 = C_1[a_1, \ldots, a_m], \ldots, x_n = C_n[a_1, \ldots, a_m]\}$, with $M_i\varphi =_E a_i$ $(1 \leq i \leq m)$ as above. Let s_i be the sort of a_i, s'_j be the sort of x_j and η a given complexity parameter.

The usual concrete semantics of φ consists in mapping every drawing of names from the set $\mathcal{E} = [\![s_1]\!]_{A_\eta} \times \cdots \times [\![s_m]\!]_{A_\eta}$ to a value in $\mathcal{F} = \mathrm{Val}_{A_\eta}(\varphi)$. Let us note $\alpha : \mathcal{E} \rightarrow \mathcal{F}$ this function, defined by:

$$\alpha(e_1, \ldots, e_m) = \Big\{x_1 = [\![C_1[y_1, \ldots, y_m]]\!]_{\{y_1=e_1,\ldots,y_m=e_m\}}, \ldots, \\ x_n = [\![C_n[y_1, \ldots, y_m]]\!]_{\{y_1=e_1,\ldots,y_m=e_m\}}\Big\}$$

where the y_i are fresh variables respectively of sort s_i, and we omit the subscript A_η for sake of clarity.

Using the M_i, we can also define a function $\beta : \mathcal{F} \rightarrow \mathcal{E}$:

$$\beta(\phi) = \Big([\![M_1]\!]_\phi, \ldots, [\![M_m]\!]_\phi\Big)$$

We note that the distribution of $[\![M_i\varphi]\!]$ equals to that of $[\![M_i]\!]_\phi$ where $\phi \xleftarrow{R} [\![\varphi]\!]$, or equivalently, of $[\![M_i]\!]_{\alpha(e_1,\ldots,e_n)}$ where $(e_1, \ldots, e_n) \xleftarrow{R} \mathcal{E}$. As $M_i\varphi =_E a_i$, (A_η) is unconditionally $=_E$-sound, and no element of $\mathcal{E}$ has probability 0, we obtain that $\beta \circ \alpha = Id_E$. Thus α is injective and yields a bijection from $\mathcal{E}$ to its image $\mathcal{G} = \alpha(\mathcal{E})$. By assumption, $\mathcal{E}$ is equipped with the uniform distribution, therefore the concrete semantics of φ is the uniform distribution on $\mathcal{G}$.

Moreover $\mathcal{G}$ satisfies:

$$\begin{aligned} \mathcal{G} &= \{\phi \in \mathcal{F} \mid \alpha(\beta(\phi)) = \phi\} \\ &= \Big\{\phi \in \mathcal{F} \mid \forall j, [\![C_j[y_1, \ldots, y_m]]\!]_{\{y_1=[\![M_1]\!]_\phi,\ldots,y_m=[\![M_i]\!]_\phi\}} = [\![x_j]\!]_\phi\Big\} \\ &= \Big\{\phi \in \mathcal{F} \mid \forall j, [\![C_j[M_1, \ldots, M_m]]\!]_\phi = [\![x_j]\!]_\phi\Big\} \end{aligned}$$

As φ is transparent, by Proposition 4, $\mathrm{eq}_E(\varphi)$ is implied by the equations $C_j[M_1, \ldots, M_m] = x_j$ and E. By unconditional $=_E$-soundness, we deduce that the values in $\mathcal{G}$ pass all the tests in $\mathrm{eq}_E(\varphi)$; in other words, $\mathcal{G} \subseteq \mathrm{Val}'_{A_\eta}(\varphi)$. Conversely, every element of $\mathrm{Val}'_{A_\eta}(\varphi)$ is trivially in $\mathcal{G}$; therefore $\mathcal{G} = \mathrm{Val}'_{A_\eta}(\varphi)$. Since $\mathcal{F}$ is equipped with uniform distribution, we obtain that the ideal semantics of φ coincides with the uniform distribution on $\mathcal{G}$, and therefore with its concrete semantics. □

A noticeable consequence of Proposition 5 is that, in the case of uniform distributions, two statically-equivalent transparent frames are always indistinguishable. (The argument is similar to that of Proposition 3.) This motivates the following definition, for the purpose of studying $\approx_E$-soundness or a converse to Proposition 3.

Definition 5. An equational theory E is *transparent* if and only if for every frame φ, there exists a (not necessarily unique) transparent frame $\overline{\varphi}$ such that $\varphi \approx_E \overline{\varphi}$.

Transparent frames and theories are related to the notion of *patterns* introduced by Abadi and Rogaway [4] and used in subsequent work [13, 12] so as to define computationally sound formal equivalences. There, messages are first mapped to patterns by replacing non-deducible subterms with boxes $\Box$. By definition, two messages are then equivalent if and only if they yield the same pattern (up to renaming of names). For example, if $\{M\}_K$ denotes the probabilistic encryption of M by a key K, the message $(\{\{K_4\}_{K_3}\}_{K_1},\, \{K_1\}_{K_2},\, K_2)$ is mapped to the pattern $(\{\Box\}_{K_1},\, \{K_1\}_{K_2},\, K_2)$. (Compare with example 1 where we have $\varphi_1 \approx_{E_{\mathsf{enc}}} \overline{\varphi_1}$.)

However, the notion of transparent frames is defined for any equational theory. Also, it might be the case that a frame corresponds to several transparent frames. For example, consider the theory of the exclusive OR (given in Section 5.1) and the frame:

$$\varphi = \{x_1 = n_1 \oplus n_2,\, x_2 = n_2 \oplus n_3,\, x_3 = n_1 \oplus n_3\}.$$

There are several transparent frames equivalent to φ, for instance $\{x_1 = n_1 \oplus n_2,\, x_2 = n_1,\, x_3 = n_2\}$, $\{x_1 = n_1,\, x_2 = n_1 \oplus n_2,\, x_3 = n_2\}$ and $\{x_1 = n_1,\, x_2 = n_2,\, x_3 = n_1 \oplus n_2\}$.

We believe that the notion of transparent frames is relevant in many theories useful in cryptography. As a matter of fact, the two theories of exclusive OR and ciphers considered in Section 5 are transparent. However, the notion of transparent frames does not subsume that of patterns, defined by Abadi and Rogaway. In particular, for the theory of probabilistic symmetric encryption, that is,

$$E_{\mathsf{senc}} = \{\mathsf{sdec}(\mathsf{senc}(x,y,z),y) = x, \quad \mathsf{sdec\_success}(\mathsf{senc}(x,y,z),y) = \mathsf{ok}\},$$

it is unclear how to associate an equivalent transparent frame to the frame $\nu n,k,r.\{x = \mathsf{senc}(n,k,r), y = k\}$, although it is arguably a pattern in the sense of Abadi and Rogaway (once cast into our syntax). The reason is that the random coin r is not deducible, but the term $\mathsf{senc}(n,k,r)$ cannot be replaced with a fresh name because of the visible equation $\mathsf{sdec\_success}(x,y) = \mathsf{ok}$. We might exclude r from being a subterm by modifying the notion of subterms (for example, in Abadi and Rogaway's work, the random factor does not appear explicitely in terms). However, this would undermine the properties of transparent frames mentioned above. Thus, we regard the notions of patterns and transparent frames as complementary.

Note that we have proved *en passant* that $\approx_E$ is decidable for transparent theories E for which $=_E$ is decidable, provided that the reduction to equivalent transparent frames is effective. Indeed, given two frames φ_1 and φ_2, we associate to each of them one of its statically equivalent transparent frame $\overline{\varphi_1}$ and $\overline{\varphi_2}$, respectively. It is then straightforward to check whether $\overline{\varphi_1}$ and $\overline{\varphi_2}$ are equivalent using the finite characterization of $\mathrm{eq}_E(\overline{\varphi_i})$ by Proposition 4.

Finally, we establish a completeness result for our soundness criterion in the cases of transparent theories.

Theorem 6. *Assume a transparent theory E. Let (A_η) be a family of computational algebras such that (A_η) has uniform distributions, is $\approx_E$-sound and unconditionally $=_E$-sound. Then the soundness criterion of Proposition 3 is satisfied: for every frame φ, $(\llbracket\varphi\rrbracket_{A_\eta}) \approx (\llbracket\varphi\rrbracket^{\mathrm{ideal}}_{A_\eta})$.*

Proof. Since E is transparent, there exists a transparent frame $\overline{\varphi}$ such that $\varphi \approx_E \overline{\varphi}$. By $\approx_E$-soundness, we deduce $(\llbracket\varphi\rrbracket_{A_\eta}) \approx (\llbracket\overline{\varphi}\rrbracket_{A_\eta})$. By Proposition 5, we have that $(\llbracket\overline{\varphi}\rrbracket_{A_\eta}) = (\llbracket\overline{\varphi}\rrbracket^{\mathrm{ideal}}_{A_\eta})$. Altogether, we conclude that $(\llbracket\varphi\rrbracket_{A_\eta}) \approx (\llbracket\varphi\rrbracket^{\mathrm{ideal}}_{A_\eta})$ since $\varphi \approx_E \overline{\varphi}$ implies $(\llbracket\overline{\varphi}\rrbracket^{\mathrm{ideal}}_{A_\eta}) = (\llbracket\varphi\rrbracket^{\mathrm{ideal}}_{A_\eta})$ as before. □

5. Examples

We now apply the framework of Sections 3 and 4 to establish two $\approx_E$-soundness results, concerning the theory of exclusive OR and that of ciphers and lists.

5.1. 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*, an infinite number of names, the infix symbol $\oplus : \mathit{Data} \times \mathit{Data} \to \mathit{Data}$ and two constants $0, 1 : \mathit{Data}$. Terms are equipped with the equational theory $E_\oplus$ generated by:

$$\begin{array}{rclcrcl} (x \oplus y) \oplus z & = & x \oplus (y \oplus z) & \qquad & x \oplus x & = & 0 \\ x \oplus y & = & y \oplus x & & x \oplus 0 & = & x \end{array}$$

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

- the concrete domain $\llbracket \mathit{Data} \rrbracket_{A_\eta}$ is the set of bit-strings of length η, $\{0,1\}^\eta$, equipped with the uniform distribution;
- $\oplus$ is interpreted by the usual XOR function over $\{0,1\}^\eta$;
- $\llbracket 0 \rrbracket_{A_\eta} = 0^\eta$ and $\llbracket 1 \rrbracket_{A_\eta} = 1^\eta$.

In this setting, statically equivalent frames enjoy an algebraic characterization. Let AC be the equational theory corresponding to the two left-hand equations for associativity and commutativity. We use the other two equations as a rewriting system $\mathcal{R}_\oplus$

$$\begin{array}{rcl} x \oplus x & \to & 0 \\ x \oplus 0 & \to & x \end{array}$$

where we allow arbitrary AC-manipulations before and after each rewriting step. It is easy to show that $\mathcal{R}_\oplus$ is AC-convergent. Specifically, a term T is in $\mathcal{R}_\oplus/AC$-normal form (or simply *normal form* in the following) if and only if each name, variable and constant 1 occur at most once in T, and 0 does not occur in T unless $T = 0$.

Let $a_1, \dots, a_n$ be distinct names. Using the rewriting system $\mathcal{R}_\oplus/AC$, every closed term T with $\text{names}(T) \subseteq \{a_1, \dots, a_n\}$ can be written $T =_{E_\oplus} \beta_0 \oplus \bigoplus_{j=1}^n \beta_j\, a_j$ where $\beta_j \in \{0,1\}$, the a_j are mutually distinct, and we use the convention $0a_j = 0$ and $1a_j = a_j$. In the following, we see $\{0,1\}$ as the two-element field $\mathbb{F}_2$; thus terms modulo $=_{E_\oplus}$ form a $\mathbb{F}_2$-vector space.

Similarly a frame φ with $\text{names}(\varphi) \subseteq \{a_1, \dots, a_n\}$ is written

$$\varphi =_{E_\oplus} \left\{ x_1 = \alpha_{1,0} \oplus \bigoplus_{j=1}^n \alpha_{1,j}\, a_j \ , \ \dots \ , \ x_m = \alpha_{m,0} \oplus \bigoplus_{j=1}^n \alpha_{m,j}\, a_j \right\}$$

where $\alpha_{i,j} \in \mathbb{F}_2$. Let us group the coefficients into a $(m+1) \times (n+1)$-matrix $\alpha = (\alpha_{i,j})$ over $\mathbb{F}_2$. Then, φ is described by the formal relation

$$\begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \dots & \alpha_{1,n} \\ \vdots & & & \vdots \\ \alpha_{m,0} & \alpha_{m,1} & \dots & \alpha_{m,n} \end{pmatrix}}_{\alpha} \cdot \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$$

We now characterize the set $\text{eq}_{E_\oplus}(\varphi)$ of equations valid in φ. Let M and N be two terms such that $\text{var}(M,N) \subseteq \text{dom}(\phi), \text{names}(M,N) \cap \text{names}(\varphi) = \emptyset$. First note that $M\varphi =_{E_\oplus} N\varphi$ if and only if $(M \oplus N)\varphi =_{E_\oplus} 0$. Therefore we only study the case where $N = 0$.

Assume M in normal form. $M\varphi =_{E_\oplus} 0$ and $\text{names}(M) \cap \text{names}(\varphi) = \emptyset$ implies $\text{names}(M) = \emptyset$. Let $M =_{AC} \beta_0 \oplus \bigoplus_{i=1}^m \beta_i\, x_i$. The condition $M\varphi =_{E_\oplus} 0$ is equivalent to the vectorial equation

$$(\beta_0, \dots, \beta_m) \cdot \alpha = 0$$

that is, $(\beta_0, \dots, \beta_m)$ belongs to the co-kernel of α, noted $\text{coker}(\alpha)$.

Finally let φ and φ' be two frames with $\text{names}(\varphi, \varphi') \subseteq \{a_1, \dots, a_n\}$ and $\text{dom}(\varphi) = \text{dom}(\varphi') = \{x_1, \dots, x_m\}$. Let α and α' be the two corresponding $(m+1) \times (n+1)$-matrices defined as above. From the previous discussion, we deduce that

$$\varphi \approx_{E_\oplus} \varphi' \ \Leftrightarrow \ \text{coker}(\alpha) = \text{coker}(\alpha')$$

that is, if we write $\text{im}(\alpha) = \{\alpha \cdot \gamma\}$ the image of α, we have by duality

$$\varphi \approx_{E_\oplus} \varphi' \ \Leftrightarrow \ \text{im}(\alpha) = \text{im}(\alpha'). \tag{1}$$

This characterization is the key point of our main result for the theory of XOR.

Theorem 7. *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.*

Proof. The unconditional $=_{E_\oplus}$-soundness is clear, hence the $\not\vdash_{E_\oplus}$-faithfulness (Proposition 1).

Let us show that (A_η) is $=_{E_\oplus}$-faithful. Assume that T_1 and T_2 are two terms such that $T_1 \neq_{E_\oplus} T_2$. This is equivalent to $T_1 \oplus T_2 \neq_{E_\oplus} 0$. Thus it is sufficient to consider the case where $T \neq 0$ is a closed term in normal form. The semantics of T is either the constant 1^η (if $T = 1$) or the uniform distribution (if $T \neq 1$) on $\{0,1\}^\eta$. Thus $\mathbb{P}\left[[\![T]\!]_{A_\eta} = 0\right]$ is negligible. Hence the $=_{E_\oplus}$-faithfulness holds and by proposition 1, so does the $\approx_{E_\oplus}$-faithfulness.

We now address the unconditional $\approx_{E_\oplus}$-soundness. Let φ be a frame, and $\alpha = (\alpha_{i,j})$ its $(m+1) \times (n+1)$-matrix associated as before. Let us see α as a $\mathbb{F}_2$-linear function from $(\mathbb{F}_2)^{n+1}$ to $(\mathbb{F}_2)^{m+1}$.

For simplicity, let us fix the order of variables in $\mathrm{dom}(\varphi)$ and assimilate the possible concrete values of φ, $\mathrm{Val}_{A_\eta}(\varphi)$, to the set $\mathcal{F} = \{1^\eta\} \times (\mathbb{F}_2)^{m\eta}$ where the first η 1-bits are added for technical reasons.

The usual concrete semantics of φ consists in drawing a random vector uniformly from $\mathcal{E} = \{1^\eta\} \times (\mathbb{F}_2)^{n\eta}$ for the value of names, and then applying a $\mathbb{F}_2$-linear function $\widehat{\alpha} : (\mathbb{F}_2)^{(n+1)\eta} \to (\mathbb{F}_2)^{(m+1)\eta}$ to it. Specifically, if we see $(\mathbb{F}_2)^{(n+1)\eta}$ as $\underbrace{{\mathbb{F}_2}^\eta \times \ldots \times {\mathbb{F}_2}^\eta}_{n+1}$ and similarly for $(\mathbb{F}_2)^{(m+1)\eta}$, the function $\widehat{\alpha}$ is defined by

$$\widehat{\alpha}\,(f_0, \ldots, f_n) = \left(\bigoplus_{j=0}^{n} \alpha_{0,j}\, f_j, \ldots, \bigoplus_{j=0}^{n} \alpha_{m,j}\, f_j \right)$$

Since $\widehat{\alpha}$ is linear, all the inverse images $\widehat{\alpha}^{-1}(\{x\})$, $x \in \mathrm{im}(\widehat{\alpha})$, have the same cardinal. Hence, the concrete semantics of φ is also the uniform distribution over $\widehat{\alpha}(\mathcal{E}) = \mathrm{im}(\widehat{\alpha}) \cap \mathcal{F}$.

Assume a second frame φ' such that $\varphi \approx_{E_\oplus} \varphi'$. Define α' and $\widehat{\alpha'}$ similarly as above. By equation 1, we have $\mathrm{im}(\alpha) = \mathrm{im}(\alpha')$.

Now, if we see $(\mathbb{F}_2)^{(m+1)\eta}$ as $\underbrace{{\mathbb{F}_2}^{m+1} \times \ldots \times {\mathbb{F}_2}^{m+1}}_{\eta}$, we may write $\widehat{\alpha} = \underbrace{\alpha \times \ldots \times \alpha}_{\eta}$ and similarly for α'. Thus,

$$\mathrm{im}(\widehat{\alpha}) = \underbrace{\mathrm{im}(\alpha) \times \ldots \times \mathrm{im}(\alpha)}_{\eta} = \underbrace{\mathrm{im}(\alpha') \times \ldots \times \mathrm{im}(\alpha')}_{\eta} = \mathrm{im}(\widehat{\alpha'})$$

which implies that φ and φ' have the same concrete semantics. Thus $E_\oplus$ is unconditionally $\approx_{E_\oplus}$-sound.

Last, we prove the unconditional $\not\vdash_{E_\oplus}$-soundness. Let φ be a frame and T a term, both in normal form, such that $\varphi \not\vdash_{E_\oplus} T$ and $\mathrm{names}(T) \subseteq \mathrm{names}(\varphi) = \{a_1, \ldots, a_n\}$. Let α be associated to φ as before and $T =_{AC} \beta_0 \oplus \bigoplus_{j=1}^{n} \beta_j\, a_j$.

Let γ be the $(m+2) \times (n+1)$-matrix obtained by augmenting α with a last row equal to $\beta = (\beta_0, \dots, \beta_n)$:

$$\gamma = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \alpha_{1,0} & \alpha_{1,1} & \dots & \alpha_{1,n} \\ \vdots & & & \vdots \\ \alpha_{m,0} & \alpha_{m,1} & \dots & \alpha_{m,n} \\ \beta_0 & \beta_1 & \dots & \beta_{n_0} \end{pmatrix}$$

Since $\varphi \not\vdash_{E_\oplus} T$, in particular there exists no M in normal form such that names$(M) = \emptyset$ and $M\varphi =_{E_\oplus} T$. In other words, β is linearly independent from the other rows in the matrix γ above.

In particular, it is independent from the first row $(1, 0, \dots, 0)$, that is, there exists $j \geq 1$ such that $\beta_j \neq 0$. We deduce that the distribution $(\xleftarrow{R} [\![T]\!]_{A_\eta})$ is the uniform one over $\{0, 1\}^\eta$, thus it is collision-free.

As for the first condition of unconditional $\not\vdash_E$-soundness, by a similar reasoning as before, we have that the concrete semantics of (φ, T) is the uniform distribution over the image of $\mathcal{E} = \{1^\eta\} \times (\mathbb{F}_2)^{n\eta}$ by $\widehat{\gamma}$ (defined similarly as $\widehat{\alpha}$ above). Let us see β a linear function from $(\mathbb{F}_2)^{n+1}$ to $\mathbb{F}_2$ and define $\widehat{\beta}$ as previously. Next we prove that the image $\widehat{\gamma}(\mathcal{E})$ is the cartesian product of the two sets $\widehat{\alpha}(\mathcal{E})$ and $\widehat{\beta}(\mathcal{E})$. It follows that the drawings for φ and T are independent.

The inclusion $\widehat{\gamma}(\mathcal{E}) \subseteq \widehat{\alpha}(\mathcal{E}) \times \widehat{\beta}(\mathcal{E})$ is trivial. As β is independent from the rows of α, there exists a vector $u \in (\mathbb{F}_2)^{n+1}$ such that $\beta(u) = 1$ and $\alpha(u) = 0$ (otherwise $\ker(\beta) \supseteq \ker(\alpha)$ implies $\beta \in \text{coim}(\beta) \subseteq \text{coim}(\alpha)$). Let $x, y \in \mathcal{E}$. We prove that there exists $z \in \mathcal{E}$ such that $\widehat{\alpha}(z) = \widehat{\alpha}(x) \in (\mathbb{F}_2)^{(m+1)\eta}$ and $\widehat{\beta}(z) = \widehat{\beta}(y) \in (\mathbb{F}_2)^\eta$.

Indeed, let us see $\mathcal{E}$ as $(\{1\} \times (\mathbb{F}_2)^n)^\eta$. Using the corresponding bases, let $x = (x_1, \dots, x_\eta)$ and $y = (y_1, \dots, y_\eta)$ with $x_i, y_i \in \{1\} \times (\mathbb{F}_2)^n$. We let $z_i = x_i + (\beta(y_i) - \beta(x_i)) \cdot u$ and $z = (z_1, \dots, z_\eta)$. Thus, $\widehat{\alpha}(z) = (\alpha(z_1), \dots, \alpha(z_\eta)) = (\alpha(x_1), \dots, \alpha(x_\eta)) = \widehat{\alpha}(x)$ and $\widehat{\beta}(z) = (\beta(z_1), \dots, \beta(z_\eta)) = (\beta(y_1), \dots, \beta(y_\eta)) = \widehat{\beta}(y)$. Besides, $\alpha(u) = 0$ implies that the first coordinate of u is 0, thus the first coordinate of each z_i is 1, that is, $z \in \mathcal{E}$. □

We conclude this section by a proof that the $E_\oplus$ is transparent as announced in Section 4.

Proposition 8. *The equational theory $E_\oplus$ is transparent.*

Proof. Indeed, let φ be frame and α be its associated $(m+1) \times (n+1)$-matrix as before. Let d be the dimension of $\text{im}(\alpha)$. There exists a $(m+1) \times d$ submatrix α' of α such that α' is injective and $\text{im}(\alpha') = \text{im}(\alpha)$ (consider a maximal independent set of columns of α). As the first column of α is independent from the others (it starts with a 1 whereas the others start with a 0), we may assume without loss of generality that the first column of α' is that of α. (In particular $d \geq 1$.)

Let $a'_1 \ldots a'_{d-1}$ be distinct names. We let φ' be the frame associated to α', described by the relation

$$\begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} = \alpha' \cdot \begin{pmatrix} 1 \\ a'_1 \\ \vdots \\ a'_{d-1} \end{pmatrix}.$$

As $\mathrm{im}(\alpha') = \mathrm{im}(\alpha)$, we have $\varphi' \approx_{E_\oplus} \varphi$. Besides, since α' is injective, there exists α'' such that $\alpha'' \cdot \alpha'$ is the identity $d \times d$-matrix. This entails that every a'_i is deducible from φ', that is, φ' is transparent. □

5.2. Symmetric, deterministic, length-preserving encryption and lists

We now detail the example of symmetric, deterministic and length-preserving encryption schemes. Such schemes, also known as *pseudo-random permutations* or *ciphers* [37], are widely used in practice, the most famous examples (for fixed-length inputs) being DES and AES.

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

$\mathsf{enc}_n, \mathsf{dec}_n$	:	$List_n \times Data \to List_n$	encryption, decryption
cons_n	:	$Data \times List_n \to List_{n+1}$	list constructor
head_n	:	$List_{n+1} \to Data$	head of a list
tail_n	:	$List_{n+1} \to List_n$	tail of a list
nil	:	$List_0$	empty list
$0, 1$	:	$Data$	constants

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

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

where x, y are variables of the appropriate sorts in each case. The effect of the last four equations is that the sort $List_0$ is degenerated in E_{sym}, that is, all terms of sort $List_0$ are equal. When oriented from left to right, the equations above form a convergent rewriting system written $\mathcal{R}$.

Notice that each 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 $\mathsf{enc}(\mathsf{cons}(k,\mathsf{nil}),k')$ instead of $\mathsf{enc}_1(\mathsf{cons}_0(k,\mathsf{nil}),k')$.

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;
- $\mathsf{enc}_n, \mathsf{dec}_n$ are implemented by a cipher for data of size $n\eta$ and keys of size η; (we discuss the required cryptographic assumptions later);
- $[\![\mathsf{nil}]\!]_{A_\eta}$ is the empty bit-string, $[\![\mathsf{cons}_n]\!]_{A_\eta}$ 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 bit-strings (of size $(n+1)\eta$) whereas $[\![\mathsf{tail}_n]\!]_{A_\eta}$ returns the last $n\eta$ digits.

We emphasize that no tags are added to messages. Tags—and in particular tags under encryption—would be harmful to the $\approx_{E_{\mathsf{sym}}}$-soundness. Indeed we expect that the formal equivalence $\nu a, b.\{x = \mathsf{enc}(a,b),\ y = b\} \approx_{E_{\mathsf{sym}}} \nu a, b, c.\{x = \mathsf{enc}(a,b),\ y = c\}$ also holds in the computational world; but this would not be the case if a is tagged before encryption. In case a was tagged before encryption, an adversary could use the tag to check the success of decrypting $\mathsf{enc}(a,b)$ with b.

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 $=_{E_{\mathsf{sym}}}$-sound (by induction on the structure of terms and equational proofs), that is, every true formal equality holds with probability 1 in the concrete world. We note that the equation $\mathsf{enc}_n(\mathsf{dec}_n(x,y),y) = x$ is satisfied because encryption by a given key is length-preserving and injective, hence also surjective.

Before studying the $\approx_{E_{\mathsf{sym}}}$-soundness, we need to characterize statically equivalent frames. Specifically, we show that this theory is transparent.

Proposition 9. *Let φ be a closed frame. There exists a transparent frame $\overline{\varphi}$ such that $\varphi \approx_{E_{\mathsf{sym}}} \overline{\varphi}$.*

The proof of Proposition 9 relies on the following Lemma 10, that is used stepwise to rewrite a frame into a transparent frame.

Lemma 10. *Let φ be a closed frame in $\mathcal{R}$-normal form. Let T be a subterm of φ of the form $T = \mathsf{enc}(U,V)$, $T = \mathsf{dec}(U,V)$, $T = \mathsf{head}(V)$ or, $T = \mathsf{tail}(V)$ and n a fresh name of the same sort than T. Assume that V is not deducible from φ, that is, $\varphi \not\vdash_{E_{\mathsf{sym}}} V$. Then we have that*

$$\varphi \approx_{E_{\mathsf{sym}}} \varphi'$$

where $\varphi' = \varphi\{T \mapsto n\}$ is obtained by replacing every *occurrence of T in φ with n.*

The proof of Lemma 10 is given in Appendix C. We prove Proposition 9 by applying this lemma repeatedly on an initial frame φ. The procedure terminates as each rewriting step decreases the total size of non-deducible subterms in the frame. Besides, the resulting frame $\overline{\varphi}$ is transparent. Indeed, by contradiction, suppose that $\overline{\varphi}$ is not transparent; define T as the father of the largest non-deducible subterm of φ; it is easy to see that T is necessarily of the form $T = \mathsf{enc}(U, V)$, $T = \mathsf{dec}(U, V)$, $T = \mathsf{head}(V)$ or $T = \mathsf{tail}(V)$ with $\varphi \not\vdash_{E_{\mathsf{sym}}} V$; thus Lemma 10 applies.

Note that for any subterm W, $\varphi \not\vdash_{E_{\mathsf{sym}}} W$ implies $\varphi\{T \mapsto n\} \not\vdash_{E_{\mathsf{sym}}} W\{T \mapsto n\}$. As a consequence, the procedure above yields a unique transparent frame $\overline{\varphi}$ (modulo renaming), no matter in which order the subterms T are substituted.

Provided that $\vdash_{E_{\mathsf{sym}}}$ is decidable[1], the above procedure for associating transparent frames to frames is effective. Thus, as noticed in Section 4.2, we obtain another proof of the decidability of $\approx_{E_{\mathsf{sym}}}$ using Proposition 4. Notice that statically equivalent transparent frames may *not* be equal modulo renaming: consider for instance $\{x = \mathsf{enc}(a, b),\, y = b\} \approx_{E_{\mathsf{sym}}} \{x = c,\, y = b\}$.

We now study the $\approx_{E_{\mathsf{sym}}}$-soundness problem under classical cryptographic assumptions. Standard assumptions on ciphers include the notions of super pseudo-random permutation (SPRP) and several notions of indistinguishability (IND-Pi-Cj, $i, j = 0, 1, 2$). 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 [37].

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 $\approx_{E_{\mathsf{sym}}}$-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. One may note that there exist operation modes which turn a fixed size block cipher realizing SPRP into a cipher which handles variable length inputs while preserving SPRP. We refer the reader to [38] for an example of such a mode and further references.

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.

[1] A classical characterization of deducibility, entailing its decidability, is detailed in Lemma 23 of Appendix C.

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

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

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 $(\mathcal{D}_{\eta,n}, \mathcal{E}_{\eta,n})$.

As in previous work [4, 13, 18, 23], we restrict frames to those with only atomic keys and no encryption cycles. Specifically, a closed frame φ *has only atomic keys* if for all subterms $\mathsf{enc}_n(u, v)$ and $\mathsf{dec}_n(u, v)$ of φ, v is a name. Given two (atomic) keys k_1 and k_2, we say that k_1 *encrypts* k_2 *in* φ, written $k_1 >_\varphi k_2$, iff there exists a subterm U of φ of the form $U = \mathsf{enc}_n(T, k_1)$ or $U = \mathsf{dec}_n(T, k_1)$ such that k_2 appears in T *not used as a key*, that is, k_2 appears in T at a position which is not the right-hand argument of a $\mathsf{enc}_{n'}$ or a $\mathsf{dec}_{n'}$. An *encryption cycle* is a tuple $k_1 \ldots k_m$ such that $k_1 >_\varphi \ldots >_\varphi k_m >_\varphi k_1$.

The effect of the condition "not used as a key" is to allow considering more terms as free of encryption cycles, for instance $\mathsf{enc}_n(\mathsf{enc}_n(a, k), k)$. This improvement is already suggested in [4].

We now state our $\approx_{E_{\mathsf{sym}}}$-soundness theorem. 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 11 ($\approx_{E_{\mathsf{sym}}}$-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 $\varphi_1 \approx_{E_{\mathsf{sym}}} \varphi_2$ then $([\![\varphi_1]\!]_{A_\eta}) \approx ([\![\varphi_2]\!]_{A_\eta})$.*

Before proving Theorem 11, we establish a computational counterpart to Lemma 10.

Lemma 12. *Let φ be a closed frame in $\mathcal{R}$-normal form, with only atomic keys and no encryption cycles. Let T be a subterm of φ of the form $T = \mathsf{enc}(U, k)$ (respectively $T = \mathsf{dec}(U, k)$), with k name of sort Data, and n a fresh name of the same sort as T. Assume that*

- *the only occurrences of k in φ are in the positions of an encryption or decryption key: $\mathsf{enc}(., k)$ or $\mathsf{dec}(., k)$;*
- *T itself does not appear under an encryption or a decryption with k;*
- *the concrete implementations for the encryption and its inverse satisfy both the ω-IND-P1-C1 assumption.*

Then we have that

$$([\![\varphi]\!]_{A_\eta}) \approx ([\![\varphi']\!]_{A_\eta})$$

where $\varphi' = \varphi\{T \mapsto n\}$ is obtained by replacing every *occurrence of T in φ with n.*

Notice that the hypothesis of Lemma 12 are stronger than its formal version, Lemma 10. For instance the encryption key k is required to be atomic; the first condition on k implies that k is not deducible from φ. Also nothing is said about head and tail symbols.

Proof (of Lemma 12). Before proving the lemma, let us consider the example of a well-formed frame $\varphi_1 = \{x_1 = \mathsf{enc}(T_1, k),\ x_2 = \mathsf{enc}(T_2, k)\}$, where k does not appear in T_1, T_2, and $T_1 \neq_{E_{\mathsf{sym}}} T_2$. This frame is statically equivalent to $\varphi_2 = \{x_1 = n_1; x_2 = n_2\}$. Our problem here is to prove that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are actually indistinguishable. It is not hard to see that this will be the case if and only if the probability that T_1 and T_2 have the same concrete value is negligible. A consequence of this phenomenon is intuitively that we need to prove Lemma 12 and—at least—a limited form of $=_{E_{\mathsf{sym}}}$-faithfulness at the same time.

Formally, let us write $|\varphi|_e$ and $|T|_e$ for the number of distinct subterms with head symbols enc or dec, occurring respectively in a frame φ and a term T. Let P_n and Q_n be the two properties:

> (P_n) Lemma 12 holds provided that $|\varphi|_e \leq n$:
> For every $\mathcal{R}$-normal, closed frame φ containing only atomic keys, no encryption cycles, and such that $|\varphi|_e \leq n$, for every maximal subterm T of φ of the form $T = \mathsf{enc}(U, k)$ or $T = \mathsf{dec}(U, k)$, for every fresh name n of the appriopriate sort, if the only occurrences of k in φ are in key positions (*i.e.* $\mathsf{enc}(., k)$ or $\mathsf{dec}(., k)$), then $([\![\varphi]\!]_{A_\eta}) \approx ([\![\varphi\{T \mapsto n\}]\!]_{A_\eta})$.
>
> (Q_n) For all $\mathcal{R}$-normal terms T_1, T_2 of the same sort such that: T_1, T_2 have only atomic keys, the frame $\varphi = \{x = T_1, y = T_2\}$ has no encryption cycles, $T_1 \neq T_2$ and $|\varphi|_e \leq n$, the probability $\mathbb{P}\left[e_1, e_2 \leftarrow [\![T_1, T_2]\!]_{A_\eta}; e_1 = e_2\right]$ is negligible.

We prove P_n and Q_n by mutual induction on n, that is, more precisely we prove the four statements: (S1) P_0, (S2) $P_{n+1} \Leftarrow Q_n$, (S3) Q_0, (S4) $Q_{n+1} \Leftarrow$ (P_{n+1} and Q_n).

(S1) P_0 is vacuously true.

(S2) $P_{n+1} \Leftarrow Q_n$. Let $T^0 = \mathsf{enc}_{n_0}(U, k)$ be a subterm of φ, k and n two names all satisfying the conditions of Lemma 12. (Naturally, the case of $T^0 = \mathsf{dec}_{n_0}(U, k)$ is similar.) Let $\varphi = \{x_1 = T^0_1, \ldots, x_n = T^0_n\}$.

Provided an adversary $\mathcal{A}$ able to distinguish $([\![\varphi]\!]_{A_\eta})$ and $([\![\varphi']\!]_{A_\eta})$, we build an adversary $\mathcal{B}$ against the ω-IND-P1-C1 assumption on encryption, described as follows:

1. for each name a of sort s appearing in φ, draw a value $\widehat{a} \xleftarrow{R} [\![s]\!]_{A_\eta}$;

2. draw a value $\widehat{a_0} \xleftarrow{R} [\![s]\!]_{A_\eta}$ for some fresh name a_0 of sort $List_{n_0}$;

3. for each x_i ($1 \leq i \leq n$) of sort s_i, compute $\widehat{T^0_i} \in [\![s_i]\!]_A$ recursively as

follows:

$$\begin{array}{rcll} \widehat{\mathsf{enc}_n(T,k)} & = & \mathcal{E}_n(\widehat{T}) \text{ if } T \neq U & \\ \widehat{\mathsf{enc}_{n_0}(U,k)} & = & \mathcal{E}^*(\widehat{U}, \widehat{a_0}) & \\ \widehat{\mathsf{dec}_n(T,k)} & = & \mathcal{D}_n(\widehat{T}) & \\ \widehat{f(T_1,\ldots,T_n)} & = & [\![f]\!]_{A_\eta}(\widehat{T_1},\ldots,\widehat{T_n}) & \text{in the remaining cases} \end{array}$$

where we have written $\mathcal{E}_n(.)$ and $\mathcal{D}_n(.)$ for the encryption and decryption oracles of the ω-IND-P1-C1 game, and $\mathcal{E}^*(\widehat{U}, \widehat{a_0})$ for the challenge ciphertext, obtained after submitting the two plaintexts $\widehat{U}$ and $\widehat{a_0}$. Since $T^0 = \mathsf{enc}_{n_0}(U,k)$ is not a subterm of an encryption or a decryption with k, we may assume that $\mathcal{E}^*(\widehat{U}, \widehat{a_0})$ is computed only once, after every call to $\mathcal{E}_n(.)$ and $\mathcal{D}_n(.)$;

4. submit the concrete frame $\{x_1 = \widehat{T_1}, \ldots, x_n = \widehat{T_n}\}$ to $\mathcal{A}$ and return the same answer.

The distribution computed by $\mathcal{B}$ and submitted to $\mathcal{A}$ equals either $([\![\varphi]\!]_{A_\eta})$ or $([\![\varphi']\!]_{A_\eta})$ depending on whichever $\mathcal{E}^*(\widehat{U}, \widehat{a_0})$ is the encryption of $\widehat{U}$, or respectively, that of $\widehat{a_0}$ (in the latter case $\mathcal{E}^*(\widehat{U}, \widehat{a_0}) = \mathcal{E}_{n_0}(\widehat{a_0})$ is simply a random number). Thus the probability that $\mathcal{B}$ guesses the right answer is the same as $\mathcal{A}$. Now it may happen that $\mathcal{B}$ does not meet the second requirement for winning the ω-IND-P1-C1 game, that is: (i) there exists a subterm $\mathsf{enc}_{n_0}(T,k)$ such that $T \neq U$ and $\widehat{T} \in \{\widehat{U}, \widehat{a_0}\}$ or (ii) there exists a subterm $\mathsf{dec}_{n_0}(T,k)$ such that $\widehat{T} = \mathcal{E}^*(\widehat{U}, \widehat{a_0})$.

For (i), the probability that $\widehat{T} = \widehat{a_0}$ is negligible by construction. Moreover, as T and $T^0 = \mathsf{enc}_{n_0}(U,k)$ are two subterms of φ and T^0 is not a subterm of T, the frame $\varphi' = \{x = T, y = U\}$ has no encryption cycles and $|\varphi'|_e < |\varphi|_e = n+1$. The induction hypothesis Q_n implies that the probability for $\widehat{T} = \widehat{U}$ is negligible.

As for (ii), if the challenge ciphertext $\mathcal{E}^*(\widehat{U}, \widehat{a_0})$ is the encryption of its second argument, that is $\mathcal{E}_{n_0}(\widehat{a_0})$, then the probability for $\widehat{T} = \mathcal{E}^*(\widehat{U}, \widehat{a_0})$ is negligible; otherwise $\mathcal{E}^*(\widehat{U}, \widehat{a_0}) = \mathcal{E}_{n_0}(\widehat{U})$. Recall that $T^0 = \mathsf{enc}_{n_0}(U,k)$ is in $\mathcal{R}$-normal form, thus $U \neq \mathsf{dec}_{n_0}(T,k)$. As T^0 and $\mathsf{dec}_{n_0}(T,k)$ are two subterms of φ and T^0 is not a subterm of $\mathsf{dec}_{n_0}(T,k)$, the frame $\varphi' = \{x = U, y = \mathsf{dec}_{n_0}(T,k)\}$ has no encryption cycles and $|\varphi'|_e < |\varphi|_e = n+1$, hence the induction hypothesis Q_n implies that the probability for $\widehat{T} = \mathcal{E}_{n_0}(\widehat{U})$ is negligible.

To simplify the case analysis of (S3) and (S4), it is convenient to introduce the following lemma:

Lemma 13. *Let T_1, T_2 be two terms of sort $List_j$. Define for each $1 \leq i \leq j$, the i-th projection of a term T of sort $List_j$, by:*

$$\pi_i(T) = \mathsf{head}(\underbrace{\mathsf{tail}(\ldots\mathsf{tail}}_{i-1 \ times}(T)))$$

Then (i) $T_1 =_{E_{\mathsf{sym}}} T_2$ *iff for all* $1 \leq i \leq j$, $\pi_i(T_1) =_{E_{\mathsf{sym}}} \pi_i(T_2)$ *and moreover (ii)* $\mathbb{P}\left[e1, e2 \leftarrow [\![T_1, T_2]\!]_{A_\eta}; e_1 = e_2\right]$ *is negligible iff for all* $1 \leq i \leq j$,

$$\mathbb{P}\left[e_1^i, e_2^i \leftarrow [\![\pi_i(T_1) \downarrow_{\mathcal{R}}, \pi_i(T_2) \downarrow_{\mathcal{R}}]\!]_{A_\eta}; e_1^i = e_2^i\right]$$

is negligible.

(The notation $T \downarrow_{\mathcal{R}}$ *stands for the* $\mathcal{R}$*-normal form of* T*.)*

Thanks to this lemma, it is sufficient to prove (S3) and (S4) for T_1 and T_2 of sort *Data* and in $\mathcal{R}$-normal form. (Indeed notice that if $\varphi = \{x = T_1, y = T_2\}$ has no encryption cycles, then $\varphi' = \{x' = \pi_i(T_1) \downarrow_{\mathcal{R}}, y' = \pi_i(T_2) \downarrow_{\mathcal{R}}\}$ has no encryption cycles and $|\varphi'|_e \leq |\varphi|_e$.)

Given the sorting system and the rewriting rules, a $\mathcal{R}$-reduced term T of sort *Data* may only be of the following forms:

1. a constant: 0 or 1,

2. a name of sort *Data*: $T = a$,

3. a projection of name of sort $List_j$: $T = \pi_i(a)$ $(1 \leq i \leq j)$,

4. a projection of a encryption/decryption of sort $List_j$: $T = \pi_i(\mathsf{enc}(U, V))$ with $U \notin \{\mathsf{dec}(T', V)\}$ or $T = \pi_i(\mathsf{dec}(U, V))$ with $U \notin \{\mathsf{enc}(T', V)\}$.

(S3) Q_0. As T_1 and T_2 contain no encryption/decryption symbol, only the cases 1–3 of the case analysis above can occur; the property follows directly.

(S4) $Q_{n+1} \Leftarrow (P_{n+1}$ and $Q_n)$. Let T_1 and T_2 be two distinct closed normal terms and $\varphi = \{x = T_1, y = T_2\}$. Assume that φ has no encryption cycles nor composed keys, and $|\varphi|_e = n + 1$.

1. If one of the two terms—say T_1— is of the form 1 (constant), 2 (name) or 3 (projection of a name). Then T_2 is of the form 4, for instance $T_2 = \pi_i(\mathsf{enc}(U, k))$ with $U \notin \{\mathsf{dec}(T', k)\}$.

    (a) If $T_1 \neq k$, by P_{n+1}, we have $([\![\varphi]\!]_{A_\eta}) \approx ([\![\{x = T_1, y = \pi_i(a)\}]\!]_{A_\eta})$ for some fresh name a. In particular, the probability for the two components x and y to be equal is negligible.

    (b) If $T_1 = k$, assume that T_1 and T_2 yields the same concrete value with significant probability. Let $List_{n_0}$ be the sort of U. We build an adversary $\mathcal{A}$ to the ω-IND-P1-C1 game as follows:

        i. for each name a of sort s appearing in T_2, draw a value $\widehat{a} \xleftarrow{R} [\![s]\!]_{A_\eta}$;

        ii. draw a value $\widehat{a_0} \xleftarrow{R} [\![s]\!]_{A_\eta}$ for some fresh name a_0 of sort $List_{n_0}$;

        iii. compute $\widehat{T_2}$ recursively as follows:

$$\begin{array}{rcll} \widehat{\mathsf{enc}_n(T, k)} & = & \mathcal{E}_n(\widehat{T}) \text{ if } T \neq U & \\ \widehat{\mathsf{enc}_{n_0}(U, k)} & = & \mathcal{E}^*(\widehat{U}, \widehat{a_0}) & \\ \widehat{\mathsf{dec}_n(T, k)} & = & \mathcal{D}_n(\widehat{T}) & \\ \widehat{f(V_1, \ldots, V_n)} & = & [\![f]\!]_{A_\eta}(\widehat{V_1}, \ldots, \widehat{V_n}) & \text{in the remaining cases} \end{array}$$

using the same conventions as before;

iv. if $\mathcal{E}_{n_0}(\widehat{U}, \widehat{T_2}) = \mathcal{E}^*(\widehat{U}, \widehat{a_0}))$, return 0, otherwise return 1.

$\mathcal{A}$ guesses the correct answer with non-negligible probability. As before, we use the property Q_n to conclude that its advantage is non-negligible.

2. Suppose $T_1 = \pi_{i_1}(\mathsf{enc}(u_1, k_1))$ and $T_2 = \pi_{i_2}(\mathsf{enc}(u_2, k_2))$ (the 3 other cases with decryption symbols are similar). As φ has no encryption cycle, we may assume for instance that k_1 is maximal for $<_\varphi$. Let T be a maximal subterm of the form $\mathsf{enc}(U, k_1)$ or $\mathsf{dec}(U, k_1)$ in φ. By P_{n+1}, we have $([\![\varphi]\!]_{A_\eta}) \approx ([\![\varphi']\!]_{A_\eta})$ where $\varphi' = \varphi\{T \mapsto a\} = \{x = T_1', y = T_2'\}$ for some fresh name a. We then apply Q_n to T_1' and T_2'. □

Proof (of Lemma 13). Point (i) is easily shown by induction on i, using the equations of E_{sym}. For (ii), notice that:

$$\mathbb{P}\left[e1, e2 \leftarrow [\![T_1, T_2]\!]_{A_\eta}; e_1 = e_2\right] \leq \sum_{i=1}^{j} \mathbb{P}\left[e_1^i, e_2^i \leftarrow [\![\pi_i(T_1), \pi_i(T_2)]\!]_{A_\eta}; e_1^i = e_2^i\right]$$

and

$$\forall i, \quad \mathbb{P}\left[e1, e2 \leftarrow [\![T_1, T_2]\!]_{A_\eta}; e_1 = e_2\right] \geq \mathbb{P}\left[e_1^i, e_2^i \leftarrow [\![\pi_i(T_1), \pi_i(T_2)]\!]_{A_\eta}; e_1^i = e_2^i\right]$$

Besides it is clear from the unconditional $=_{E_{\mathsf{sym}}}$-soundness, that for any T_1, T_2:

$$\mathbb{P}\left[e1, e2 \leftarrow [\![T_1, T_2]\!]_{A_\eta}; e_1 = e_2\right] = \mathbb{P}\left[e1, e2 \leftarrow [\![T_1 \downarrow_{\mathcal{R}}, T_2 \downarrow_{\mathcal{R}}]\!]_{A_\eta}; e_1 = e_2\right]$$

□

Proof (of Theorem 11). Thanks to the (unconditional) $=_{E_{\mathsf{sym}}}$-soundness, it is enough to prove the property on frames in $\mathcal{R}$-normal form.

We begin by proving the following lemma:

Lemma 14. *Assume that the concrete implementations for the encryption and its inverse satisfy both the ω-IND-$P1$-$C1$ assumption. For every well-formed $\mathcal{R}$-normal frame φ, $([\![\varphi]\!]_{A_\eta}) \approx ([\![\overline{\varphi}]\!]_{A_\eta})$ where $\overline{\varphi}$ is the transparent frame associated to φ following the algorithmic proof of Proposition 9 (this transparent frame is uniquely defined modulo renaming of names.).*

Now recall that by Proposition 5 and since $\varphi \approx \overline{\varphi}$, we have:

$$[\![\overline{\varphi}]\!]_{A_\eta} = [\![\overline{\varphi}]\!]_{A_\eta}^{\text{ideal}} = [\![\varphi]\!]_{A_\eta}^{\text{ideal}}$$

Therefore the soundness criterion holds for well-formed $\mathcal{R}$-normal frames and we conclude by Proposition 3. □

Notice that the use of the ideal semantics could not be easily avoided as two statically equivalent transparent frames may not be equal modulo renaming of bound names.

Proof (of Lemma 14). We prove the property by induction on the number m of encryptions and decryptions by non-deducible keys in φ.

If $m = 0$, by the well-formedness condition, φ is already a transparent frame.

Suppose that $m > 0$. As φ has no encryption cycle, we choose a non-deducible (atomic) key k appearing in φ, such that k is maximal for the encryption relation $>_\varphi$.

As k is not deducible, is maximal for $>_\varphi$ and φ contains no head and tail symbols, the only occurrences of k in φ are as encryption or decryption keys. Let T be a maximal subterm of φ of the form $T = \mathsf{enc}(U, k)$ or $T = \mathsf{dec}(U, k)$. We apply Lemma 12 on φ and T and conclude by induction hypothesis on the obtained frame φ'. □

Note on the cryptographic assumptions.. Cryptographic assumptions of Theorem 11 may appear strong compared to existing work on passive adversaries [4, 13]. This seems unavoidable when we allow frames to contain both encryption and decryption symbols.

In the case where the two frames to be compared contain no decryption symbols, our proofs are easily adapted to work when the encryption scheme is ω-IND-P1-C0 only, where ω-IND-P1-C0 is defined similarly to ω-IND-P1-C1 except that the adversary has no access to the decryption oracle. Such an assumption is realizable in practice using a variable-input-length cipher [39, 38].

Finally, it should be possible to recover the classical assumption IND-P1-C1 by modeling the ECB mode (Electronic Code Book). Consider two new symbols $\mathsf{enc} : \mathit{Data} \times \mathit{Data} \to \mathit{Data}$ and $\mathsf{dec} : \mathit{Data} \times \mathit{Data} \to \mathit{Data}$, and define the symbols enc_n and dec_n (formally and concretely) recursively by

$$\begin{array}{rcl} \mathsf{enc}_{n+1}(x, y) & = & \mathsf{cons}_n(\mathsf{enc}(\mathsf{head}_n(x), y), \mathsf{enc}_n(\mathsf{tail}_n(x), y)) \quad \text{and} \\ \mathsf{dec}_{n+1}(x, y) & = & \mathsf{cons}_n(\mathsf{dec}(\mathsf{head}_n(x), y), \mathsf{dec}_n(\mathsf{tail}_n(x), y)) \end{array}$$

together with the equations

$$\begin{array}{rcl} \mathsf{dec}(\mathsf{enc}(x, y), y) & = & x \\ \mathsf{enc}(\mathsf{dec}(x, y), y) & = & y \end{array}$$

Define well-formed frames as those of which the normal forms contain no encryption cycles. Then, similar techniques can be applied to show that $\approx_{E_{\mathsf{sym}}}$-soundness holds for well-formed frames as soon as the implementations for enc and dec are both IND-P1-C1, or equivalently [37], enc is SPRP.

Note on the well-formedness assumptions.. We may also note that it is possible to slightly relax the assumptions of well-formedness of frames. In particular we could allow encryption cycles on deducible keys and for instance allow the frame $\{x = \mathsf{enc}(k_1, k_2),\ y = \mathsf{enc}(k_2, k_1),\ z = k_1\}$ which is currently discarded. As these extensions are not essential for our results we prefer to avoid unnecessary clutter and keep the definitions simple.

6. Conclusion and future work

In this paper we developed a general framework for relating formal and computational models of security protocols in the presence of a passive attacker. These are the first results on abstract models allowing arbitrary equational theories. We define the soundness and faithfulness of cryptographic implementations with respect to abstract models. We also provide a soundness criterion which is not only sufficient but also necessary for many theories. Finally, we provide new soundness results for the exclusive OR and a theory of ciphers and lists.

A direction for further work is to study the soundness of other theories. An interesting case is the combination of the two theories considered in this paper, that is modeling the exclusive OR, ciphers and lists. Another interesting open problem is to generalize the notion of transparent frames so as to include probabilistic encryption, while retaining the essential properties of transparent frames. Finally, an ambitious extension is to consider the case of an active attacker in presence of general equational theories.

Acknowledgments.. We would like to thank the anonymous reviewers for their helpful suggestions. This work was partially supported by the ACI JC 9005, the ARA SSIA FormaCrypt and the ANR SESUR AVOTÉ.

References

[1] M. Baudet, V. Cortier, S. Kremer, Computationally sound implementations of equational theories against passive adversaries, in: Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), Vol. 3580 of LNCS, Springer, 2005, pp. 652–663.

[2] D. Dolev, A. C. Yao, On the security of public key protocols, IEEE Transactions on Information Theory IT-29 (12) (1983) 198–208.

[3] S. Goldwasser, S. Micali, Probabilistic encryption, Journal of Computer and System Sciences 28 (1984) 270–299.

[4] M. Abadi, 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), Vol. 1872 of LNCS, 2000, pp. 3–22.

[5] H. Comon, V. Shmatikov, Is it possible to decide whether a cryptographic protocol is secure or not?, Journal of Telecommunications and Information Technology (4/2002) 5–15.

[6] V. Cortier, S. Delaune, P. Lafourcade, A survey of algebraic properties used in cryptographic protocols, Journal of Computer Security 14 (1) (2006) 1–43.

[7] M. Abadi, C. Fournet, Mobile values, new names, and secure communications, in: Proc. 28th Annual ACM Symposium on Principles of Programming Languages (POPL'01), 2001, pp. 104–115.

[8] R. Corin, J. Doumen, S. Etalle, Analysing password protocol security against off-line dictionary attacks, in: Proc. 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP'04), Vol. 121 of ENTCS, 2005, pp. 47–63.

[9] M. Baudet, Deciding security of protocols against off-line guessing attacks, in: Proc. 12th ACM Conference on Computer and Communications Security (CCS'05), ACM Press, 2005, pp. 16–25.

[10] M. Abadi, V. Cortier, Deciding knowledge in security protocols under equational theories, in: Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04), Vol. 3142 of LNCS, 2004, pp. 46–58.

[11] B. Blanchet, Automatic proof of strong secrecy for security protocols, in: Proc. 25th IEEE Symposium on Security and Privacy (SSP'04), 2004, pp. 86–100.

[12] P. Adão, G. Bana, A. Scedrov, Computational and information-theoretic soundness and completeness of formal encryption, in: Proc. 18th IEEE Computer Security Foundations Workshop (CSFW'05), 2005, pp. 170–184.

[13] D. Micciancio, B. Warinschi, Completeness theorems for the Abadi-Rogaway logic of encrypted expressions, Journal of Computer Security 12 (1) (2004) 99–129.

[14] P. Laud, Computationally secure information flow, Ph.D. thesis, Universität des Saarlandes (2002).

[15] P. Laud, R. Corin, Sound computational interpretation of formal encryption with composed keys, in: Proc. 6th International Conference on Information Security and Cryptology (ICISC'03), Vol. 2971 of LNCS, 2004, pp. 55–66.

[16] P. Adão, J. Herzog, G. Bana, A. Scedrov, Soundness of formal encryption in the presence of key-cycles, in: Proc. 10th European Symposium on Research in Computer Security (ESORICS'05), Vol. 3679 of LNCS, 2005, pp. 374–396.

[17] M. Backes, B. Pfitzmann, M. Waidner, A composable cryptographic library with nested operations, in: Proc. 10th ACM Conference on Computer and Communications Security (CCS'03), ACM Press, 2003, pp. 220–230.

[18] M. Backes, B. Pfitzmann, Symmetric encryption in a simulatable Dolev-Yao style cryptographic library, in: Proc. 17th IEEE Computer Science Foundations Workshop (CSFW'04), 2004, pp. 204–218.

[19] M. Backes, B. Pfitzmann, M. Waidner, Symmetric authentication within simulatable cryptographic library, in: Proc. 8th European Symposium on Research in Computer Security (ESORICS'03), LNCS, 2003, pp. 271–290.

[20] V. Cortier, B. Warinschi, Computationally sound, automated proofs for security protocols, in: Proc. 14th European Symposium on Programming (ESOP'05), Vol. 3444 of LNCS, 2005, pp. 157–171.

[21] R. Janvier, Y. Lakhnech, L. Mazaré, Completing the picture: Soundness of formal encryption in the presence of active adversaries, in: Proc. 14th European Symposium on Programming (ESOP'05), Vol. 3444 of LNCS, 2005, pp. 172–185.

[22] R. Canetti, J. Herzog, Universally composable symbolic analysis of mutual authentication and key-exchange protocols (extended abstract), in: Proc. 3rd Theory of Cryptography Conference (TCC'06), Vol. 3876 of LNCS, 2006, pp. 380–403.

[23] P. Laud, Symmetric encryption in automatic analyses for confidentiality against active adversaries, in: Proc. IEEE Symposium on Security and Privacy (SSP'04), 2004, pp. 71–85.

[24] A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, M. Turuani, Probabilistic Polynomial-time Semantics for a Protocol Security Logic, in: Proc. 32nd International Colloquium on Automata, Languages and Programming, ICALP, Vol. 3580 of LNCS, Springer, 2005, pp. 16–29, lisboa, Portugal.

[25] B. Blanchet, A computationally sound mechanized prover for security protocols, in: IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 2006, pp. 140–154.

[26] M. Backes, B. Pfitzmann, Limits of the cryptographic realization of dolev-yao-style xor, in: Proc. 10th European Symposium on Research in Computer Security (ESORICS'05), Vol. 3679 of LNCS, 2005, pp. 336–354.

[27] M. Abadi, B. Warinschi, Password-based encryption analyzed, in: Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), Vol. 3580 of LNCS, 2005, pp. 664–676.

[28] M. Abadi, M. Baudet, 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), Vol. 3921 of LNCS, 2006, pp. 398–412.

[29] G. Bana, P. Mohassel, T. Stegers, The computational soundness of formal indistinguishability and static equivalence, in: Proc. 11th Asian Computing Science Conference (ASIAN'06), Vol. 4435 of LNCS, Springer, 2006, pp. 182–196.

[30] S. Kremer, L. Mazaré, Adaptive soundness of static equivalence, in: Proc. 12th European Symposium on Research in Computer Security (ESORICS'07), Vol. 4734 of LNCS, Springer, 2007, pp. 610–625.

[31] H. Comon-Lundh, V. Cortier, Computational soundness of observational equivalence, in: Proc. 15th ACM Conference on Computer and Communications Security (CCS'08), ACM Press, 2008, pp. 109–118.

[32] M. Abadi, C. Fournet, Mobile values, new names, and secure communication, in: Proc. of the 28th ACM Symposium on Principles of Programming Languages (POPL'01), 2001, pp. 104–115.

[33] S. Hohenberger, The cryptographic impact of groups with infeasible inversion, Master's thesis, MIT (2003).

[34] R. L. Rivest, On the notion of pseudo-free groups, in: Proc. 1st Theory of Cryptography Conference (TCC'04), Vol. 2951 of LNCS, 2004, pp. 505–521.

[35] S. Goldwasser, M. Bellare, Lecture notes on cryptography (2008).

[36] D. Micciancio, The RSA group is pseudo-free, in: Advances in Cryptology – Proc. EUROCRYPT '05, Vol. 3494 of LNCS, 2005, pp. 387–403.

[37] D. H. Phan, D. Pointcheval, About the security of ciphers (semantic security and pseudo-random permutations), in: Proc. Selected Areas in Cryptography (SAC'04), Vol. 3357 of LNCS, 2004, pp. 185–200.

[38] S. Halevi, Invertible universal hashing and the TET encryption mode, in: Advances in Cryptology – Proc. CRYPTO '2007, Vol. 4622 of LNCS, 2007, pp. 412–429.

[39] M. Bellare, P. Rogaway, On the construction of variable-input-length ciphers, in: Proc. 6th Workshop on Fast Software Encryption (FSE'99), Vol. 1636 of LNCS, 1999, pp. 231–244.

[40] E. Contejean, C. Marché, B. Monate, X. Urbain, The CiME Rewrite Tool, `http://cime.lri.fr` (2000).

A. General results on static equivalence

We prove here some general properties of static equivalence concerning free symbols. We first establish a useful interpolation lemma.

Given a term $U = f(U_1, \ldots, U_n)$ where f is a free symbol (see Section 2.1) and a name a of the same sort as U, the *cutting function* $\mathsf{cut}_{U,a}$ is defined recursively as follows: $\mathsf{cut}_{U,a}(u) = u$ if u is a variable or a name, and

$$\mathsf{cut}_{U,a}(g(T_1, \ldots, T_k)) = \begin{cases} a \quad \text{if } g = f,\ k = n \text{ and } \forall 1 \leq i \leq n,\ U_i =_E T_i \\ g(\mathsf{cut}_{U,a}(T_1), \ldots, \mathsf{cut}_{U,a}(T_k)) \quad \text{otherwise} \end{cases}$$

Thus, the effect of function $\mathsf{cut}_{U,a}(T)$ is to substitute some (but not all) subterms of T equal to U modulo E with a.

Lemma 15. *Let $U = f(U_1, \ldots, U_n)$ be a term such that f is a free symbol. Let a be a name of the same sort as U. For any two terms M and N,*

$$M =_E N \quad \textit{implies} \quad \mathsf{cut}_{U,a}(M) =_E \mathsf{cut}_{U,a}(N).$$

Proof. By Birkhoff's theorem, $M =_E N$ means that there exist $n \geq 0$ and $M_0, \ldots, M_n$ such that $M = M_0 \leftrightarrow_E M_1 \leftrightarrow_E M_n = N$ where $\leftrightarrow_E$ denotes one step of rewriting along one equation in (the generating set of) E, oriented in either direction.

To prove the property by induction on n, it suffices to consider the case $n = 1$. More precisely, assume that there exists an equation $l = r$ in E, a position p and a substitution θ such that $M|_p = l\theta$ and $N = M[r\theta]_p$. By definition of free symbols, we may assume that f does not occur in l and r. We consider two cases depending on whether the cutting function $\mathsf{cut}_{U,a}$ cuts a subterm above p or not.

- Either there exists a proper prefix p' of p such that $M|_{p'} = f(T_1, \ldots, T_k)$ and for all i, $U_i =_E T_i$. We consider the smallest p' that satisfies this property. Thus $p = p' \cdot i \cdot p''$ and $N = M[f(T_1, \ldots, T_i[r\theta]_{p''}, \ldots, T_n)_{p'}]$. Both terms $f(T_1, \ldots, T_k)$ and $f(T_1, \ldots, T_i[r\theta]_{p''}, \ldots, T_n)$ are substituted with a, thus $\mathsf{cut}_{U,a}(M) = \mathsf{cut}_{U,a}(N)$.

- Or no such cutting position p' is a proper prefix of p. This means that $\mathsf{cut}_{U,a}(M[x]_p) = \mathsf{cut}_{U,a}(N[x]_p)$ and $\mathsf{cut}_{U,a}(M) = \mathsf{cut}_{U,a}(M[x]_p)[\mathsf{cut}_{U,a}(l\theta)]_p$, where x is a fresh variable. Moreover, $\mathsf{cut}_{U,a}(l\theta) = l\mathsf{cut}_{U,a}(\theta)$ and $\mathsf{cut}_{U,a}(r\theta) = r\mathsf{cut}_{U,a}(\theta)$ since f is free. We deduce

$$\begin{array}{rcl} \mathsf{cut}_{U,a}(M) & = & \mathsf{cut}_{U,a}(M[x]_p)[\mathsf{cut}_{U,a}(l\theta)]_p \\ & = & \mathsf{cut}_{U,a}(N[x]_p)[l\mathsf{cut}_{U,a}(\theta)]_p \\ & =_E & \mathsf{cut}_{U,a}(N[x]_p)[r\mathsf{cut}_{U,a}(\theta)]_p \\ & = & \mathsf{cut}_{U,a}(N) \end{array}$$

Using this lemma, we establish two simple properties of free symbols.

Corollary 16. *Let f be a free symbol and $f(T_1, \ldots, T_n)$ a term of a non-degenerated type τ.*

1. *For every $U_1, \ldots, U_n$ of the appropriate sort,*

$$f(T_1, \ldots T_n) =_E f(U_1, \ldots, U_n) \quad \textit{iff} \quad \forall i,\ T_i =_E U_i.$$

2. *Let U be a term of sort τ such that f does not appear in U. Then*

$$f(T_1, \ldots, T_n) \neq_E U.$$

Proof.

1. The right-to-left implication is trivial. Let $T = f(T_1, \dots T_n)$ and $U = f(U_1, \dots, U_n)$. By contradiction, assume that there exists an i such that $T_i \neq_E U_i$. Let a_1, a_2 be two fresh names of sort τ. We apply Lemma 15 on the equation $T =_E U$ successively with cut_{T,a_1} and cut_{U',a_2} where $U' = \mathsf{cut}_{T,a_1}(U) = f(\mathsf{cut}_{T,a_1}(U_1), \dots \mathsf{cut}_{T,a_1}(U_n))$. We obtain $a_1 =_E a_2$, hence τ is degenerated; contradiction.

2. Assume $f(T_1, \dots, T_n) =_E U$. Then by Lemma 15, since f does not occur in U, we obtain $a =_E U$ for some fresh name a, hence τ is degenerated; contradiction.

We are now ready to prove our propositions.

Proposition 17. *Let T_1, T_2 be two terms of sort s such that $T_1 \neq_E T_2$. Assume a free symbol $\mathsf{h}_s : s \times Key \to Hash$ such that the sort Key is not degenerated. Consider the frame $\varphi_1 = \{x_1 = \mathsf{h}_s(T_1, k),\ x_2 = \mathsf{h}_s(T_2, k)\}$ where k is a fresh name. Let $\varphi_2 = \{x_1 = n,\ x_2 = n'\}$ where n, n' are two distinct fresh names of sort Hash. Then we have $\varphi_1 \approx_E \varphi_2$.*

Proof. Let M and N be two terms such that $\mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi)$ and $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi) = \emptyset$.

Assume $M\varphi_2 =_E N\varphi_2$. Let θ be the substitution $\{n \mapsto \mathsf{h}_s(T_1, k), n' \mapsto \mathsf{h}_s(T_2, k)\}$. Since the equational theory E is stable by substitution of names, we have $M\varphi_2\theta =_E N\varphi_2\theta$, hence, $M\varphi_1 =_E N\varphi_1$ as n, n' are fresh names.

Conversely, assume $M\varphi_1 =_E N\varphi_1$. Let $U_1 = \mathsf{h}_s(T_1, k)$. By Lemma 15, we have $\mathsf{cut}_{U_1,n}(M\varphi_1) =_E \mathsf{cut}_{U_1,n}(N\varphi_1)$. Since k does not appear in M nor N, by Corollary 16, it holds that $\mathsf{cut}_{U_1,n}(M\varphi_1) = M\mathsf{cut}_{U_1,n}(\varphi_1)$ and $\mathsf{cut}_{U_1,n}(N\varphi_1) = N\mathsf{cut}_{U_1,n}(\varphi_1)$. Now, using $T_1 \neq_E T_2$, we prove $\mathsf{cut}_{U_1,n}(\varphi_1) = \{x_1 = n,\ x_2 = \mathsf{h}_s(T_2, k)\}$. Indeed, we have $\mathsf{cut}_{U_1,n}(\mathsf{h}_s(T_2, k)) = \mathsf{h}_s(\mathsf{cut}_{U_1,n}(T_2), k)$ since $T_1 \neq_E T_2$. Besides, as k does not appear in T_2, by Corollary 16, we have $\mathsf{cut}_{U_1,n}(T_2) = T_2$. Similarly, by applying $\mathsf{cut}_{U_2,n'}$ with $U_2 = \mathsf{h}_s(T_2, k)$, we obtain

$$M\mathsf{cut}_{U_2,n'}(\mathsf{cut}_{U_1,n}(\varphi_1)) =_E N\mathsf{cut}_{U_2,n'}(\mathsf{cut}_{U_1,n}(\varphi_1)),$$

that is, $M\varphi_2 =_E N\varphi_2$. □

Proposition 18. *Let φ be a frame and T a term of sort s. Assume a free symbol $\mathsf{h}_s : s \times Key \to Hash$ such that the sort Key is not degenerated. Let $\varphi_1 = \varphi \cup \{x = \mathsf{h}_s(T, k), y = k\}$ and $\varphi_2 = \varphi \cup \{x = n, y = k\}$ where x, y are fresh variables, k is a fresh name of sort Key, n is a fresh name of sort Hash. If $\varphi \not\vdash_E T$, then $\varphi_1 \approx_E \varphi_2$.*

Proof. Let M and N be two terms such that $\mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi)$ and $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi) = \emptyset$. We prove that $M\varphi_2 =_E N\varphi_2$ implies $M\varphi_1 =_E N\varphi_1$ similarly as for Proposition 17.

Conversely, assume $M\varphi_1 =_E N\varphi_1$. Let $U = \mathsf{h}_s(T, k)$. By Lemma 15, we have $\mathsf{cut}_{U,n}(M\varphi_1) =_E \mathsf{cut}_{U,n}(N\varphi_1)$. Let us prove that $\mathsf{cut}_{U,n}(M\varphi_1) = M\mathsf{cut}_{U,n}(\varphi_1)$.

Indeed, otherwise, there exists a subterm M_1 of M such that M_1 is not a variable and $M_1\varphi_1 = \mathsf{h}_s(T', T'')$ with $T' =_E T$ and $T'' =_E k$. Since M_1 is not a variable, M_1 is of the form $M_1 = \mathsf{h}_s(M_1', M_1'')$ with $M_1'\varphi_1 = T' =_E T$, which implies that T is deducible; contradiction.

We deduce that $\mathsf{cut}_{U,n}(M\varphi_1) = M\mathsf{cut}_{U,n}(\varphi_1)$, and similarly $\mathsf{cut}_{U,n}(N\varphi_1) = N\mathsf{cut}_{U,n}(\varphi_1)$. Thus $M\mathsf{cut}_{U,n}(\varphi_1) =_E N\mathsf{cut}_{U,n}(\varphi_1)$. By Corollary 16, as k does not appear in φ, we have that $\mathsf{cut}_{U,n}(\varphi) = \varphi$, hence $\mathsf{cut}_{U,n}(\varphi_1) = \varphi_2$ and $M\varphi_2 =_E N\varphi_2$. □

Proposition 19. *Let T_1, T_2 be two terms of sort s such that $T_1 =_E T_2$. Assume a free symbol $\mathsf{h}_s : s \times Key \to Hash$ such that Key is not degenerated. Let $\varphi = \{x_1 = \mathsf{h}_s(T_1, k),\ x_2 = \mathsf{h}_s(T_2, k)\}$. Then, $\varphi \approx_E \{x_1 = n,\ x_2 = n\}$ where n is a fresh name of sort Hash.*

Proof. Let M and N be two terms such that $\mathrm{var}(M, N) \subseteq \mathrm{dom}(\varphi)$ and $\mathrm{names}(M, N) \cap \mathrm{names}(\varphi) = \emptyset$. We prove that $M\varphi_2 =_E N\varphi_2$ implies $M\varphi_1 =_E N\varphi_1$ similarly as for Proposition 17.

Conversely, assume $M\varphi_1 =_E N\varphi_1$. Let $U = \mathsf{h}_s(T_1, k)$. By Lemma 15, we have $\mathsf{cut}_{U,n}(M\varphi_1) =_E \mathsf{cut}_{U,n}(N\varphi_1)$. Since k does not appear in M nor N, by Corollary 16, we have $\mathsf{cut}_{U,n}(M\varphi_1) = M\mathsf{cut}_{U,n}(\varphi_1)$ and $\mathsf{cut}_{U,n}(N\varphi_1) = N\mathsf{cut}_{U,n}(\varphi_1)$. Now, since $T_1 =_E T_2$, we obtain $\mathsf{cut}_{U,n}(\varphi_1) = \{x_1 = n,\ x_2 = n\} = \varphi_2$. Thus we have $M\varphi_2 =_E N\varphi_2$. □

B. Static equivalence in groups

We establish some properties of static equivalence in the equational theory of Abelian groups E_{G} defined in Section 3.2. For this purpose we characterize equivalence classes in E_{G} by a representation lemma.

Let $\mathcal{X}_A$ ($\mathcal{X}_G$ and $\mathcal{X}_{Hash}$ respectively) be the set of variables of sort A (G and *Hash* respectively). Let $\mathcal{N}_A$ ($\mathcal{N}_G$ and $\mathcal{N}_{Hash}$ respectively) be the set of names of sort A (G and *Hash* respectively). Let AC be the equational theory corresponding to the subset of equations from E_{G}, modeling the associativity and commutativity of the three operators $\cdot$, $+$ and $*$.

We call *unitary monomial of sort A* a function $\beta : \mathcal{X}_A \cup \mathcal{N}_A \to \mathbb{N}$ almost everywhere zero, *i*.e., except for a finite number of entries. Such a function β can be considered as a term of sort A (modulo AC):

$$\beta =_{AC} \prod_{a \in \mathcal{N}_A,\ \beta(a) \neq 0} a^{\beta(a)} \cdot \prod_{u \in \mathcal{X}_A,\ \beta(u) \neq 0} u^{\beta(u)}$$

where empty products are considered to be the term 1_A, and $a^{\beta(a)}$ ($\beta(a) \neq 0$) denotes the term $\underbrace{a \cdot \ldots \cdot a}_{\beta(a) \text{ times}}$. We denote $\mathcal{M}_A$ the set of all unitary monomials of sort A.

A *canonical form of sort A* is a function $\alpha : \mathcal{M}_A \rightarrow \mathbb{Z}$ almost everywhere zero. We consider such a function α as a term of sort A (modulo AC):

$$\alpha =_{AC} \sum_{\beta \in \mathcal{M}_A,\ \alpha(\beta) \neq 0} \alpha(\beta) \cdot \beta$$

where empty sums are considered to be the term 0_A, and integers are naturally represented as 0_A, $1_A + \ldots + 1_A$ or $-(1_A + \ldots + 1_A)$ of sort A.

A *canonical form of sort G* is a function γ, mapping terms in $\mathcal{X}_N \cup \mathcal{N}_N$ to canonical forms of sort A, almost everywhere zero, *i.e.*, the function evaluates to the constant 0 except for a finite number of entries. We consider a canonical form γ to be a term of sort G (modulo AC):

$$\gamma =_{AC} \prod_{g \in \mathcal{N}_G,\ \gamma(g) \neq 0} g^{\gamma(g)} * \prod_{x \in \mathcal{X}_G,\ \gamma(x) \neq 0} x^{\gamma(x)}$$

where empty products are considered to be equal to 1_G.

A *canonical form of sort Hash*, denoted ι, is either a variable of sort *Hash* : $\iota = z \in \mathcal{X}_{Hash}$, a name of sort *Hash* : $\iota = h \in \mathcal{N}_{Hash}$ or a canonical form γ of sort G considered to be a term $\iota = \mathsf{h}(\gamma)$.

Lemma 20. *For any term T of sort A (G, Hash, respectively), there exists a unique canonical form α_T (γ_T, ι_T, respectively) such that*

$$T =_{E_{\mathsf{G}}} \alpha_T$$

($T =_{E_{\mathsf{G}}} \gamma_T$, $T =_{E_{\mathsf{G}}} \iota_T$, respectively).

Proof (Sketch). We show the existence of a canonical form of a term T by induction on the structure of T. For instance, given $T = T_1 * T_2$, and two canonical forms α_{T_1} and α_{T_2}, we obtain the canonical form of T by rearranging the product $\alpha_{T_1} * \alpha_{T_2}$ modulo E_{G} (and if necessary the induction hypthesis is also used on the exponents). To show the uniqueness of the normal form, it is sufficient to show that whenever two canonical terms are equal as terms modulo E_{G}, they are also equal "mathematically". Formally this is established by studying the AC normal form of each canonical form with respect to the following AC-convergent rewriting system.

$$\begin{array}{rcl@{\qquad\qquad}rcl}
u + 0_A & \rightarrow & u & x * 1_G & \rightarrow & x \\
u + (-u) & \rightarrow & 0_A & (x^u)^v & \rightarrow & x^{(u \cdot v)} \\
u \cdot 1_A & \rightarrow & u & x^u * x^v & \rightarrow & x^{u+v} \\
(u+v) \cdot w & \rightarrow & u \cdot w + v \cdot w & x^{1_A} & \rightarrow & x \\
u \cdot 0_A & \rightarrow & 0_A & x^{0_A} & \rightarrow & 1_G \\
-(u+v) & \rightarrow & (-u) + (-v) & (x * y)^u & \rightarrow & x^u * y^u \\
(-u) \cdot v & \rightarrow & -(u \cdot v) & x * x & \rightarrow & x^{(1_A + 1_A)} \\
-(-u) & \rightarrow & u & x * x^u & \rightarrow & x^{u+1_A} \\
-0_A & \rightarrow & 0_A & (1_G)^u & \rightarrow & 1_G
\end{array}$$

This rewriting system has been obtained by orienting and completing the equations generating E_{G}, except AC, using the tool Cime [40]. □

Proposition 21. *Let* $\varphi_1 = \nu g, a, b.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^{a\cdot b}\}$ *and* $\varphi_2 = \nu g, a, b, c.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^c\}$*. We have that* $\varphi_1 \approx_{E_{\mathsf{G}}} \varphi_2$*.*

Proof. Let M, N be two terms of the same sort such that $\mathrm{var}(M,N) \subseteq \mathrm{dom}(\varphi_1)$ and $\mathrm{names}(M,N) \cap \mathrm{names}(\varphi_1,\varphi_2) = \emptyset$.

Assume $M\varphi_2 =_{E_{\mathsf{G}}} N\varphi_2$. Let θ be the substitution $\{c \mapsto a \cdot b\}$. Since the equational theory E is stable by substitution of names, we have $M\varphi_2\theta =_E N\varphi_2\theta$, that is, $M\varphi_1 =_{E_{\mathsf{G}}} N\varphi_1$ since $c \notin \mathrm{names}(M,N)$.

Conversely, assume $M\varphi_1 =_{E_{\mathsf{G}}} N\varphi_1$. If M and N are of sort A, then $\mathrm{var}(M,N) = \emptyset$ and hence $M\varphi_2 = M\varphi_1 =_E N\varphi_1 = N\varphi_2$.

Otherwise, M and N are of sort G. As $M\varphi_1 =_{E_{\mathsf{G}}} N\varphi_1$ is equivalent to $M\varphi_1 * (N\varphi_1)^{-1_A} = 1_G$, we suppose that $N = 1_G$.

As $\mathrm{var}(M) \subseteq \mathrm{dom}(\varphi_1)$ and $\mathrm{names}(M) \cap \mathrm{names}(\varphi_1,\varphi_2) = \emptyset$, the canonical form γ of M is of the form

$$M =_{E_{\mathsf{G}}} \prod_{g' \neq g} {g'}^{\gamma(g')} * x_1^{\gamma(x_1)} * \ldots * x_4^{\gamma(x_4)}$$

where $\gamma(g')$ and $\gamma(x_i)$ represent closed terms with disjoint names $\{a, b, c\}$. Hence, we have that

$$M\varphi_1 =_{E_{\mathsf{G}}} \prod_{g' \neq g} {g'}^{\gamma(g')} * g^{\gamma(x_1) + \gamma(x_2)\cdot a + \gamma(x_3)\cdot b + \gamma(x_4) a\cdot b} =_{E_{\mathsf{G}}} 1_G$$

and we conclude that for any i, $\gamma(x_i) = 0_A$ and for any g', $\gamma(g') = 0_A$, *i.e.*, $M = 1_G$. □

Proposition 22. *Let the frame* $\varphi_1 = \nu g, a.\{x_1 = g^a, x_2 = a, x_3 = \mathsf{h}(g)\}$ *and the frame* $\varphi_2 = \nu g, a, h.\{x_1 = g^a, x_2 = a, x_3 = h\}$*. We have that* $\varphi_1 \approx_{E_{\mathsf{G}}} \varphi_2$*.*

Proof. Let M, N be two terms such that $\mathrm{var}(M,N) \subseteq \mathrm{dom}(\varphi_1)$ and $\mathrm{names}(M,N) \cap \mathrm{names}(\varphi_1,\varphi_2) = \emptyset$.

Assume $M\varphi_2 =_E N\varphi_2$. Let θ be the substitution $\{h \mapsto \mathsf{h}(g)\}$. Since the equational theory E is stable by substitution of names, we have $M\varphi_2\theta =_E N\varphi_2\theta$, hence, as $h \notin \mathrm{names}(M,N)$, $M\varphi_1 =_E N\varphi_1$.

Conversely, assume that $M\varphi_1 =_{E_{\mathsf{G}}} N\varphi_1$. If M and N are of sort A or G, then $\mathrm{var}(M,N) \subseteq \{x_1, x_2\}$ and hence $M\varphi_2 = M\varphi_1 =_E N\varphi_1 = N\varphi_2$.

Otherwise, M and N are of sort *Hash*. We suppose that $M = x_3$ and $N = \mathsf{h}(N')$ where $\mathrm{var}(N') \subseteq \{x_1, x_2\}$ (other cases are trivial). As h is a free symbol, by Corollary 16, $M\varphi_1 =_{E_{\mathsf{G}}} N\varphi_1$ is equivalent to $N'\varphi_1 =_{E_{\mathsf{G}}} g$.

Given that $\mathrm{var}(N') \subseteq \{x_1, x_2\}$ and $\mathrm{names}(N') \cap \mathrm{names}(\varphi_1,\varphi_2) = \emptyset$, the canonical form γ of N' is of the form

$$N' =_{E_{\mathsf{G}}} \prod_{g' \neq g} {g'}^{\gamma(g')} * x_1^{\gamma(x_1)}$$

where $\gamma(g')$ and $\gamma(x_1)$ are terms that have no variable other than x_2 and do not contain a. Hence we have

$$N'\varphi_1 =_{E_{\mathsf{G}}} \prod_{g' \neq g} {g'}^{\gamma(g')\{x_2 \mapsto a\}} * g^{\gamma(x_1)\cdot a}$$

which contradicts $N'\varphi_1 =_{E_{\mathsf{G}}} g$. □

C. Static equivalence in ciphers and lists

Before proving Lemma 10, we first introduce a handy lemma to characterize deducible terms.

Lemma 23. *Let $\varphi = \nu\tilde{n}.\sigma$ be a closed frame in $\mathcal{R}$-normal form and T a term in $\mathcal{R}$-normal form. If $\varphi \vdash_{E_{\mathsf{sym}}} T$ then $T = C[T_1, \ldots, T_k]$ where the T_i are deducible subterms of φ and C is a context that does not contain private names that is* names$(C) \cap \tilde{n} = \emptyset$.

Proof. By definition, $\varphi \vdash_{E_{\mathsf{sym}}} T$ if and only if there exists a term M such that names$(M) \cap$ names$(\varphi) = \emptyset$ and $M\varphi =_{E_{\mathsf{sym}}} T$, that is, $M\varphi \to^*_{\mathcal{R}} T$. We prove Lemma 23 by induction on the size of M. The base case $M = x_i$ is trivial.

If $M = f(M_1, \ldots, M_k)$. We only consider the case where $M = \mathsf{dec}(M_1, M_2)$ since the other cases are similar. We have $M_1 \to^*_{\mathcal{R}} T_1$ and $M_2 \to^*_{\mathcal{R}} T_2$. By applying the induction hypothesis to M_1 and M_2, we obtain that $T_1 = C_1[T'_1, \ldots, T'_k]$ and $T_2 = C_2[T'_1, \ldots, T'_k]$ where the T'_i are deducible subterms of φ and C_1, C_2 are contexts that do not contain names. We have $M\varphi \to^*_{\mathcal{R}} \mathsf{dec}(T_1, T_2)$. Either $\mathsf{dec}(T_1, T_2)$ is in $\mathcal{R}$-normal form. In that case and by convergence of $\mathcal{R}$, we have $T = \mathsf{dec}(T_1, T_2)$, hence the result. Or $\mathsf{dec}(T_1, T_2)$ is not in $\mathcal{R}$-normal form. By convergence, we have $\mathsf{dec}(T_1, T_2) \to_{\mathcal{R}} T$. Since T_1 and T_2 are already in normal form, we must have $T_1 = \mathsf{enc}(T'_1, T_2)$ and $T = T'_1$. Either $C_1 = \mathsf{enc}(C'_1, C''_1)$ and we have $T = C'_1[T'_1, \ldots, T'_k]$. Or $C_1 = \_$, which means that T_1 is a deducible subterm of φ. We deduce that T is a deducible subterm of φ, hence the result. □

We can now start the proof of Lemma 10.

Proof. In what follows, we say that a term or a context is *public* if it does not contain the names occurring in φ. Since $\varphi = \varphi'\{n \mapsto T\}$ and E_{sym} is stable by substitutions of names, we have $\mathrm{eq}_{E_{\mathsf{sym}}}(\varphi') \subseteq \mathrm{eq}_{E_{\mathsf{sym}}}(\varphi)$. To prove $\mathrm{eq}_{E_{\mathsf{sym}}}(\varphi) \subseteq \mathrm{eq}_{E_{\mathsf{sym}}}(\varphi')$, we introduce the following lemma. We set θ to be $\{n \mapsto T\}$. Let $n_1, \ldots, n_p$ be the names occurring in φ'.

Lemma 24. *Let C_1 be a context such that we have $\varphi' \vdash_{E_{\mathsf{sym}}} C_1[n_1, \ldots, n_p]$ and $C_1[n_1, \ldots, n_p]\theta \to_{\mathcal{R}} T$. Then there exists a public context C_2 such that $C_1 \to_{\mathcal{R}} C_2$ and $T = C_2[n_1, \ldots, n_p]\theta$.*

The lemma is proved by inspection of the rules of $\mathcal{R}$. The reduction occurs at some position p: the reduction $C_1[n_1, \ldots, n_p]|_p\theta \to_{\mathcal{R}} T$ occurs in head. Let $C'_1[n_1, \ldots, n_p] = C_1[n_1, \ldots, n_p]|_p$ If C'_1 is itself an instance of the left-hand-side of a rule of $\mathcal{R}$, than we clearly have that $C'_1 \to_{\mathcal{R}} C'_2$ such that $T = C_2[n_1, \ldots, n_p]\theta$, where C_2 is obtained from C_1 by replacing C'_1 with C'_2 at position p. If C'_1 is not an instance of the left-hand-side of a rule of $\mathcal{R}$ and since T is already in $\mathcal{R}$-normal form, there are only four possibilities for $C'_1[n_1, \ldots, n_p]$.

- $C'_1[n_1, \ldots, n_p] = \mathsf{enc}(n_i, C''_1[n_1, \ldots, n_p])$. It must be the case that $n_i = n$, T is of the form $\mathsf{dec}(U, V)$ and $V = C''_1[n_1, \ldots, n_p]$. From Lemma 23 and since $\varphi' \vdash_{E_{\mathsf{sym}}} C_1[n_1, \ldots, n_p]$, either $C'_1[n_1, \ldots, n_p]$ is subterm of φ' or n_i and $C''_1[n_1, \ldots, n_p]$ are deducible. In both cases, we obtain a contradiction. Indeed, if $C'_1[n_1, \ldots, n_p]$ is subterm of φ' then $C'_1[n_1, \ldots, n_p]\theta = \mathsf{enc}(\mathsf{dec}(U, V), n_j)$ is a subterm of φ, which contradicts that φ is in normal form. If n_i and $C''_1[n_1, \ldots, n_p]$ are deducible then this contradicts $\varphi \not\vdash_{E_{\mathsf{sym}}} V$.

- $C'_1[n_1, \ldots, n_p] = \mathsf{dec}(n_i, n_j)$. This case is very similar to the previous one.

- $C'_1[n_1, \ldots, n_p] = \mathsf{cons}(n_i, C''_1[n_1, \ldots, n_p])$. It must be the case that $n_i = n$, T is of the form $\mathsf{head}(V)$ and $C''_1[n_1, \ldots, n_p] = \mathsf{tail}(V)$. From Lemma 23 and since $\varphi' \vdash_{E_{\mathsf{sym}}} C_1[n_1, \ldots, n_p]$, either $C'_1[n_1, \ldots, n_p]$ is subterm of φ' or n_i and $C''_1[n_1, \ldots, n_p]$ are deducible. As previously, in both cases, we obtain a contradiction. if $C'_1[n_1, \ldots, n_p]$ is subterm of φ' then $C'_1[n_1, \ldots, n_p]\theta = \mathsf{cons}(\mathsf{head}(V), \mathsf{tail}(V))$ is a subterm of φ, which contradicts that φ is in normal form. If n_i and $C''_1[n_1, \ldots, n_p]$ are deducible then both n and $\mathsf{tail}(V)$ are deducible in φ', which means that both $\mathsf{head}(V)$ and $\mathsf{tail}(v)$ are deducible in φ, thus V is deducible in φ, contradiction.

- $C'_1[n_1, \ldots, n_p] = \mathsf{cons}(C''_1[n_1, \ldots, n_p], n_i)$. This case is very similar to the previous one.

Now, let $(M = N) \in \mathrm{eq}_{E_{\mathsf{sym}}}(\varphi)$ and let us show that $(M = N) \in \mathrm{eq}_{E_{\mathsf{sym}}}(\varphi')$. We have $M\varphi =_{E_{\mathsf{sym}}} N\varphi$, that is, $M\varphi'\theta =_{E_{\mathsf{sym}}} N\varphi'\theta$. By convergence of $\mathcal{R}$, there exists a term T such that $M\varphi'\theta \to^*_{\mathcal{R}} T$ and $N\varphi'\theta \to^*_{\mathcal{R}} T$. By applying repeatedly Lemma 24, we obtain that $M\varphi' \to^*_{\mathcal{R}} T_1$ such that $T = T_1\theta$ and $N\varphi' \to^*_{\mathcal{R}} T_2$ such that $T = T_2\theta$. Assume that we have proved that $T_1 = T_2$. Then we have $M\varphi' =_{E_{\mathsf{sym}}} N\varphi'$, that is, $(M = N) \in \mathrm{eq}_{E_{\mathsf{sym}}}(\varphi')$, which concludes the proof. It remains for us to prove the following lemma.

Lemma 25. *Let T_1 and T_2 be two terms such that each T_i is either deducible from φ', that is, $\varphi' \vdash_{E_{\mathsf{sym}}} T_i$, or T_i is a subterm of φ'. Then $T_1\theta = T_2\theta$ implies $T_1 = T_2$.*

The lemma is proved by induction on the sum of the size of T_1 and T_2. First notice that, by Lemma 23, any subterm T' of one of the T_i verifies that T' is deducible from φ' or T' is a subterm of φ'.

- The base case is trivial.

- If none of T_1 or T_2 is n: $T_1 = f(T'_1, \ldots, T'_k)$ and $T_2 = f(T''_1, \ldots, T''_k)$. We must have $T'_i\theta = T''_i\theta$ for every $1 \leq i \leq k$. By applying the induction hypothesis, we obtain $T'_i = T''_i$ thus $T_1 = T_2$.

- The most difficult case is when $T_1 = n$ and $T_2 = f(T'_1, \ldots, T'_k)$. We first notice that since $n\theta = f(T'_1, \ldots, T'_k)\theta$, n cannot occur in T_2, thus $T_2 = T_2\theta$. Either T_2 is a subterm of φ', which is impossible by construction of φ' or T_2 deducible. Since T_2 is not a subterm of φ' and applying again Lemma 23, we get that the immediate subterms of T_2 are deducible in φ' (thus in φ), which contradicts the choice of T. □

Formal Indistinguishability extended to the Random Oracle Model

Cristian Ene, Yassine Lakhnech and Van Chan Ngo [*]

Université Grenoble 1, CNRS,VERIMAG

Abstract. Several generic constructions for transforming one-way functions to asymmetric encryption schemes have been proposed. One-way functions only guarantee the weak secrecy of their arguments. That is, given the image by a one-way function of a random value, an adversary has only negligible probability to compute this random value. Encryption schemes must guarantee a stronger secrecy notion. They must be at least resistant against indistinguishability-attacks under chosen plaintext text (IND-CPA). Most practical constructions have been proved in the random oracle model (ROM for short). Such computational proofs turn out to be complex and error prone. Bana et al. have introduced *Formal Indistinguishability Relations (FIR)*, as an abstraction of computational indistinguishability. In this paper, we extend the notion of FIR to cope with the ROM on one hand and adaptive adversaries on the other hand. Indeed, when dealing with hash functions in the ROM and one-way functions, it is important to correctly abstract the notion of weak secrecy. Moreover, one needs to extend frames to include adversaries in order to capture security notions as IND-CPA. To fix these problems, we consider pairs of formal indistinguishability relations and *formal non-derivability relations.* We provide a general framework along with general theorems, that ensure soundness of our approach and then we use our new framework to verify several examples of encryption schemes among which the construction of Bellare Rogaway and Hashed ElGamal.

1 Introduction

Our day-to-day lives increasingly depend upon information and our ability to manipulate it securely. That is, in a way that prevents malicious elements to subvert the available information for their own benefits. This requires solutions based on *provably correct* cryptographic systems (e.g., primitives and protocols). There are two main frameworks for analyzing cryptographic systems; the *symbolic framework*, originating from the work of Dolev and Yao [16], and the *computational approach*, growing out of the work of [18]. A significant amount of effort has been made in order to link both approaches and profit from the advantages of each of them. Indeed, while the symbolic approach is more amenable to automated proof methods, the computation approach can be more realistic.

[*] Grenoble, email:name@imag.fr This work has been partially supported by the ANR projects SCALP, AVOTE and SFINCS

In their seminal paper [1] Abadi and Rogaway investigate the link between the symbolic model on one hand and the computational model on the other hand. More precisely, they introduce an equivalence relation on terms and prove that equivalent terms correspond to indistinguishable distributions ensembles, when interpreted in the computational model. The work of Abadi and Rogaway has been extended to active adversaries and various cryptographic primitives in e.g. [21, 20, 14, 19]. An other line of work, also considering active adversaries is followed by Backes, Pfitzmann and Waidner using *reactive simulatability* [5, 4] and Canetti [12, 13] using *universal composability.*

Related works. A recently emerging branch of relating symbolic and computational models for passive adversaries is based on *static equivalence* from π-calculus [3], induced by an *equational theory.* Equational theories provide a framework to specify algebraic properties of the underlying signature, and hence, symbolic computations in a similar way as for abstract data types. That is, for a fixed equational theory, a term describes a computation in the symbolic model. Thus, an adversary can distinguish two terms, if he is able to come up with two computations that yield the same result when applied to one term but different results when applied to the other term. Such a pair of terms is called a *test.* This idea can be extended to *frames*, which roughly speaking are tuples of terms. Thus, a *static equivalence* relation is fully determined by the underlying equational theory, as two frames are *statically equivalent*, if there is no test that separates them. In [8] Baudet, Cortier and Kremer study soundness and faithfulness of static equivalence for general equational theories and use their framework to prove soundness of exclusive or as well as certain symmetric encryptions. Abadi et al. [2] use static equivalence to analyze guessing attacks.

Bana, Mohassel and Stegers [7] argue that even though static equivalence works well to obtain soundness results for the equational theories mentioned above, it does not work well in other important cases. Consider for instance the Decisional Diffie Hellman assumption (DDH for short) that states that the tuples (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c), are indistinguishable for randomly sampled a, b, c. It does not seem to be obvious to come up with an equational theory for group exponentiation such that the induced static equivalence includes this pair of tuples without including others whose computational indistinguishability is not proved to be a consequence of the DDH assumption. The static equivalence induced by the equational theory for group exponentiation proposed in [8] includes the pair (g, g^a, g^b, g^{a^2b}) and (g, g^a, g^b, g^c). It is unknown whether the computational indistinguishability of these two distributions can be proved under the DDH assumption. Therefore, Bana et al. propose an alternative approach to build symbolic indistinguishability relations and introduce *formal indistinguishability relations (FIR).* A FIR is defined as a closure of an initial set of equivalent frames with respect to simple operations which correspond to steps in proofs by reduction. This leads to a flexible symbolic equivalence relation. FIR has nice properties. In order to prove soundness of a FIR it is enough to prove soundness of the initial set of equivalences. Moreover, static equivalence is one instance of a FIR. Bana et al. show that it is possible to come up with a FIR whose soundness is equivalent to the DDH assumption.

The techniques introduced in this paper, borrow and generalize to arbitrary equational theories some ideas from [15]. In [15] the authors provide a specialized Hoare-like logic to reason about encryption schemes in the random oracle model, and apply their logic to prove IND-CPA of several schemes, including the generic encryption scheme of Bellare and Rogaway [10].

Contributions. In this paper, we extend Bana et al.'s approach by introducing a notion of symbolic equivalence that allows us to prove security of encryption schemes symbolically. More specifically, we would like to be able to treat generic encryption schemes that transform one-way functions to IND-CPA secure encryption schemes. Therefore, three problems need to be solved. First, we need to cope with one-way functions. This is a case where the static equivalence does not seem to be appropriate. Indeed, let f be a one-way function, that is, a function that is easy to compute but difficult to invert. It does not seem easy to come with a set of equations that capture the one-wayness of such a function. Consider the term $f(a|b)$, where $|$ is bit-string concatenation. We know that we cannot easily compute $a|b$ given $f(a|b)$ for uniformly sampled a and b. However, nothing prevents us from being able to compute a for instance. Introducing equations that allow us to compute a from $f(a|b)$, e.g., $g(f(a|b)) = a$, may exclude some one-way functions and does not solve the problem. For instance, nothing prevents us from computing a prefix of b, a prefix of the prefix, etc ... The second problem that needs to be solved is related to the fact that almost all practical provably secure encryption schemes are analyzed in the random oracle model (ROM for short). ROM is an idealized model in which hash functions are randomly sampled functions. In this model, adversaries have oracle access to these functions. An important property is that if an adversary is unable to compute the value of an expression a and if $H(a)$ has not been leaked then $H(a)$ looks like a uniformly sampled value. Thus, we need to be able to symbolically prove that a value of a given expression a cannot be computed by any adversary. This is sometimes called *weak secrecy* in contrast to indistinguishability based secrecy. To cope with this problem, our notion of symbolic indistinguishability comes along with a *non-derivability* symbolic relation. Thus in our approach, we start from an initial pair of a non-derivability relation and a frame equivalence relation. Then, we provide rules that define a closure of this pair of relations in the spirit of Bana et al.'s work. Also in our case, soundness of the obtained relations can be checked by checking soundness of the initial relations. The third problem is related to the fact that security notions for encryption schemes such IND-CPA and real-or-random indistinguishability of cipher-text under chosen plaintext involve *active* adversaries. Indeed, these security definitions correspond to two-phase games, where the adversary first computes a value, then a challenge is produced, then the adversary tries to solve the challenge. Static equivalence and FIR (as defined in [7]) consider only passive adversaries. To solve this problem we consider frames that include variables that correspond to adversaries. As frames are finite terms, we only have finitely many such variables. This is the reason why we only have a degenerate form of active adversaries which is enough to treat security of encryption schemes and digital signature, for instance. The closure rules we propose in our framework are designed with the objective of minimizing the initial relations which depend on the underlying cryptographic primitives and assumptions. We illustrate the framework by considering security proofs of the construction of Bellare and Rogaway [10] and Hash El Gamal [6].

Outline of the paper. In Section 2, we introduce the symbolic model used for describing generic asymmetric encryption schemes. In Section 3, we describe the computational framework and give definitions that relate the two models. In Section 4, we introduce our definition of formal indistinguishability relation and formal non-derivability relation. We also present our method for proving IND-CPA security. In Section 5, we illustrate our framework: we prove the constructions of Bellare and Rogaway [10], Hash El Gamal [6], and the encryption scheme proposed by Pointcheval in [24]. Finally, in Section 7 we conclude.

2 Symbolic semantics

A *signature* $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{H})$ consists of a countable infinite set of *sorts* $\mathcal{S} = \{s, s_1, ...\}$, a finite set of *function symbols*, $\mathcal{F} = \{f, f_1, ...\}$, and a finite set of *oracle symbols*, $\mathcal{H} = \{g, h, h_1, ...\}$ together with arities of the form $ar(f)$ or $ar(h) = s_1 \times ... \times s_k \to s, k \geq 0$. Symbols in $\mathcal{F}$ that take $k = 0$ as arguments are called *constants*. We suppose that there are three pairwise disjoint countable sets $\mathcal{N}$, $\mathcal{X}$ and $\mathcal{P}$. $\mathcal{N}$ is the set of names, $\mathcal{X}$ is the set of first-order variables, and $\mathcal{P}$ is the set of second order variables. We assume that both names and variables are sorted, that is, to each name or variable u, a sort $\mathbf{s}$ is assigned; we use $\mathbf{s}(u)$ for the sot of u. Variables $p \in \mathcal{P}$ have arities $ar(p) = \mathbf{s}_1 \times ... \times \mathbf{s}_k \to \mathbf{s}$.

A renaming is a bijection $\tau : \mathcal{N} \to \mathcal{N}$ such that $\mathbf{s}(a) = \mathbf{s}(\tau(a))$. As usual, we extend the notation $\mathbf{s}(T)$ to denote the sort of a term T. Terms of sort $\mathbf{s}$ are defined by the grammar:

$$\begin{array}{lll} T ::= & x & \textit{variable } x \textit{ of sort } \mathbf{s} \\ & |n & \textit{name } n \textit{ of sort } \mathbf{s} \\ & |p(T_1, \ldots, T_k) & \textit{variable } p \textit{ of arity } \mathbf{s}(T_1) \times ... \times \mathbf{s}(T_k) \to \mathbf{s} \\ & |f(T_1, \ldots, T_k) & \textit{application of } f \in \mathcal{F} \textit{ with arity } \mathbf{s}(T_1) \times ... \times \mathbf{s}(T_k) \to \mathbf{s} \\ & |h(T_1, \ldots, T_k) & \textit{call of } h \in \mathcal{H} \textit{ with arity } \mathbf{s}(T_1) \times ... \times \mathbf{s}(T_k) \to \mathbf{s} \end{array}$$

We use $fn(T)$, $pvar(T)$ and $var(T)$ for the set of free names, the set of p-variables and the set of variables that occur in the term T, respectively. Meta-variables u, v, w range over names and variables. We use $st(T)$ for the set of subterms of T, defined in the usual way: $st(u) \overset{def}{=} \{u\}$ if u is a name or a variable, and $st(l(T_1, \ldots, T_k)) \overset{def}{=} \{l(T_1, \ldots, T_k)\} \bigcup_{i \in \{1,\ldots k\}} st(T_i)$, if $l \in \mathcal{F} \cup \mathcal{H} \cup \mathcal{P}$. A term T is closed if it does not have any free variables (but it may contain p-variables), that means $var(T) = \emptyset$. The set of terms is denoted by $\mathbf{T}$.

Symbols in $\mathcal{F}$ are intended to model cryptographic primitives, symbols in $\mathcal{H}$ are intended to model cryptographic oracles (in particular, hash functions in the ROM model), and names in $\mathcal{N}$ are used to model secrets, i.e. concretely random numbers. Variables $p \in \mathcal{P}$ are intended to model queries and challenges made by adversaries (and can depend on previous queries).

Definition 1 (Substitution). *A substitution $\sigma = \{x_1 = T_1, ..., x_n = T_n\}$ is a mapping from variables to terms whose domain $dom(\sigma) = \{x_1, ..., x_n\}$ is finite and such that $\sigma(x) \neq x$, for each x in the domain.*

A substitution as above is *well-sorted* if x_i and T_i have the same sort for each i, and there is no circular dependence $x_{i_2} \in var(T_{i_1})$, $x_{i_3} \in var(T_{i_2})$, $\ldots$, $x_{i_1} \in var(T_{i_k})$. The application of a substitution σ to a term T is written as $\sigma(T) = T\sigma$. This definition is lifted in a standard way to the application of a substitution to set of terms or substitutions. The *normal form* σ^* of a well-sorted substitution σ is the iterative composition of σ with itself until it remains unchanged : $\sigma^* = (\ldots((\sigma)\sigma)\ldots)\sigma$. For example, if $\sigma = \{x_1 = a, x_2 = f(b, x_1), x_3 = g(x_1, x_2)\}$, then $\sigma^* = \{x_1 = a, x_2 = f(b, a), x_3 = g(a, f(b, a)\}$. A substitution is *closed* if all terms (of its normal form) T_i are closed. We let $var(\sigma) = \cup_i var(T_i)$, $pvar(\sigma) = \cup_i pvar(T_i)$, $n(\sigma) = \cup_i fn(T_i)$, and extend the notations $pvar(.)$, $var(.)$, $n(.)$ and $st(.)$ to tuples and set of terms in the obvious way.

The abstract semantics of symbols is described by an equational theory E, that is an equivalence (denoted as $=_E$) which is stable with respect to application of contexts and well-sorted substitutions of variables.

Definition 2 (Equational Theory.). *An equational theory for a given signature is an equivalence relation $E \subseteq \mathcal{T} \times \mathcal{T}$ (written as $=_E$ in infix notation) on the set of terms such that*
1) $T_1 =_E T_2$ implies $T_1\sigma =_E T_2\sigma$ for every substitution σ;
2) $T_1 =_E T_2$ implies $T\{x = T_1\} =_E T\{x = T_2\}$ for every term T and every variable x;
3) $T_1 =_E T_2$ implies $\tau(T_1) =_E \tau(T_2)$ for every renaming τ.

Frames ([3]) represent sequences of messages observed by an adversary. Formally:

Definition 3 (Frame). *A frame is an expression of the form $\phi = \nu\widetilde{n}.\sigma$ where σ is a well-sorted substitution, and $\widetilde{n}$ is $n(\sigma)$, the set of all names occurring in σ. By abuse of notation we also use $n(\phi)$ for $\widetilde{n}$, the set of names bounded in the frame ϕ. We note $fv(\phi) \overset{def}{=} var(\sigma) \setminus dom(\sigma)$ the set of free variables of ϕ.*

The novelty of our definition of frames consists in permitting adversaries to interact with frames using p-variables. This is necessary to be able to cope with adaptive adversaries. We note the set of frames by $\mathbf{F}$.

The normal form ϕ^* of a frame $\phi = \nu\widetilde{n}.\sigma$ is the frame $\phi^* = \nu\widetilde{n}.\sigma^*$. From now on, we tacitly identify substitutions and frames with their normal form. Next, we define composition of frames. Let $\phi = \nu\widetilde{n}.\{x_1 = T_1, ..., x_n = T_n\}$ and $\phi' = \nu\widetilde{n'}.\sigma$ be frames with $\widetilde{n} \cap \widetilde{n'} = \emptyset$. Then, $\phi\phi'$ denotes the frame $\nu(\widetilde{n} \cup \widetilde{n'}).\{x_1 = T_1\sigma, ..., x_n = T_n\sigma\}$.

Definition 4 (Equational equivalence). *Let ϕ and ϕ' be two frames such that $\phi^* = \nu\widetilde{n}.\sigma$ and $\phi'^* = \nu\widetilde{n}.\sigma'$ with $\sigma = \{x_1 = T_1, ..., x_n = T_n\}$ and $\sigma' = \{x_1 = T'_1, ..., x_n = T'_n\}$. Given the equational theory E, we say that ϕ and ϕ' are equationally equivalent written $\phi =_E \phi'$, if and only if $T_i\sigma =_E T'_i\sigma'$ for all i.*

3 Computational Semantics

3.1 Distributions and indistinguishability

Let us note $\eta \in \mathrm{N}$ the security parameter. We are interested in analyzing generic schemes for asymmetric encryption in the *random oracle model* [17, 10]. We write $h \overset{r}{\leftarrow} \Omega$ to denote that h is randomly chosen from the set of functions with appropriate domain (depending on η). By abuse of notation, for a list $\boldsymbol{H} = h_1, \cdots, h_m$ of hash functions, we write $\boldsymbol{H} \overset{r}{\leftarrow} \Omega$ instead of the sequence $h_1 \overset{r}{\leftarrow} \Omega, \ldots, h_m \overset{r}{\leftarrow} \Omega$. We fix a finite set $\mathcal{H} = \{h_1, \ldots, h_n\}$ of hash functions. A *distribution ensemble* is a countable sequence of distributions $\{X_\eta\}_{\eta \in \mathrm{N}}$. We only consider distribution ensembles that can be constructed in polynomial time by probabilistic algorithms that have oracle access to $\mathcal{O} = \mathcal{H}$. Given two distribution ensembles $X = \{X_\eta\}_{\eta \in \mathrm{N}}$ and $X' = \{X'_\eta\}_{\eta \in \mathrm{N}}$, an algorithm $\mathcal{A}$ and $\eta \in \mathrm{N}$, the *advantage* of $\mathcal{A}$ in distinguishing X_η and X'_η is defined by:

$\mathsf{Adv}(\mathcal{A}, \eta, X, X') = \mathsf{Pr}[x \overset{r}{\leftarrow} X_\eta : \mathcal{A}^{\mathcal{O}}(\eta, x) = 1] - \mathsf{Pr}[x \overset{r}{\leftarrow} X'_\eta : \mathcal{A}^{\mathcal{O}}(\eta, x) = 1]$.

Then, two distribution ensembles X and X' are called *indistinguishable* (denoted by $X \sim X'$) if for any probabilistic polynomial-time algorithm $\mathcal{A}$, the advantage $\mathsf{Adv}(\mathcal{A}, \eta, X, X')$ is negligible as a function of η, that is, for any $n > 0$, it become eventually smaller than η^{-n} as η tends to infinity.

3.2 Frames as distributions

We now give terms and frames a computational semantics parameterized by a computable implementation of the primitives in ROM. Provided a set of sorts $\mathcal{S}$ and a set of symbols $\mathcal{F}$, a *computational algebra* $A = (\mathcal{S}, \mathcal{F})$ consists of

- a sequence of non-empty finite set of bit strings $[\![s]\!]_A = \{[\![s]\!]_{A,\eta}\}_{\eta \in \mathrm{N}}$ with $[\![s]\!]_{A,\eta} \subseteq \{0,1\}^*$ for each sort $s \in S$. For simplicity of the presentation, we assume that all sorts are large domains, whose cardinalities are exponential in the security parameter η;

- a sequence of polynomial time computable functions $[\![f]\!]_A = \{[\![f]\!]_{A,\eta}\}_{\eta \in \mathrm{N}}$ with $[\![f]\!]_{A,\eta} : [\![s_1]\!]_{A,\eta} \times ... \times [\![s_k]\!]_{A,\eta} \to [\![s]\!]_{A,\eta}$ for each $f \in \mathcal{F}$ with $ar(f) = s_1 \times ... \times s_k \to s$;

- a polynomial time computable congruence $=_{A,\eta,s}$ for each sort s, in order to check the equality of elements in $[\![s]\!]_{A,\eta}$ (the same element may be represented by different bit strings). By congruence, we mean a reflexive, symmetric, and transitive relation such that $e_1 =_{A,s_1,\eta} e'_1, ..., e_k =_{A,s_k,\eta} e'_k \Rightarrow [\![f]\!]_{A,\eta}(e_1, ..., e_k) =_{A,s,\eta} [\![f]\!]_{A,\eta}(e'_1, ..., e'_k)$ (we usually omit s,η and A and write $=$ for $=_{A,s,\eta}$);

- a polynomial time procedure to draw random elements from $[\![s]\!]_{A,\eta}$; we denote such a drawing by $x \leftarrow^R [\![s]\!]_{A,\eta}$; for simplicity, in this paper we suppose that all these drawing follow a uniform distribution.

From now on we assume a fixed computational algebra $(\mathcal{S}, \mathcal{F})$, and a fixed η, and for simplicity we omit the indices A,s and η. For lack of space, we use *ppt* to stand for probabilistic polynomial-time. Given $\mathcal{H}$ a fixed set of hash functions, and $(\mathcal{A}_i)_{i \in I}$ a fixed set of ppt functions (can be seen as a ppt adversary $\mathcal{A}^{\mathcal{O}}$ taking an additional input i), we associate to each frame $\phi = \nu\widetilde{n}.\{x_1 = T_1, \ldots, x_k = T_k\}$ a sequence of distributions $[\![\phi]\!]_{\mathcal{H},\mathcal{A}}$ computed as follows:

- for each name n of sort s appearing in $\widetilde{n}$, draw a value $\hat{n} \overset{r}{\leftarrow} [\![s]\!]$;

- for each variable $x_i (1 \leq i \leq k)$ of sort s_i, compute $\hat{T_i} \in [\![s_i]\!]$ recursively on the structure of terms: $\hat{x_i} = \hat{T_i}$;

- for each call $h_i(T'_1, \ldots, T'_m)$ compute recursively on the structure of terms: $h_i(\widehat{T'_1, \ldots, T'_m}) = h_i(\hat{T'_1}, \ldots, \hat{T'_m})$;

- for each call $f(T'_1, \ldots, T'_m)$ compute recursively on the structure of terms: $f(\widehat{T'_1, \ldots, T'_m}) = [\![f]\!](\hat{T'_1}, \ldots, \hat{T'_m})$;

- for each call $p_i(T'_1, \ldots, T'_m)$ compute recursively on the structure of terms and draw a value $p_i(\widehat{T'_1, \ldots, T'_m}) \overset{r}{\leftarrow} \mathcal{A}^{\mathcal{O}}(i, \hat{T'_1}, \ldots, \hat{T'_m})$;

- return the value $\hat{\phi} = \{x_1 = \hat{T_1}, \ldots, x_k = \hat{T_k}\}$.

Such $\phi = \{x_1 = bse_1, \ldots, x_n = bse_n\}$ with $bse_i \in [\![s_i]\!]$ are called *concrete frames*. We extend the notation $[\![.]\!]$ to (sets of) closed terms in the obvious way.

Now the concrete semantics of a frame ϕ with respect to an adversary $\mathcal{A}$, is given by the following sequence of distributions (one for each implicit η):

$[\![\phi]\!]_{\mathcal{A}} = \left[\mathcal{H} \overset{r}{\leftarrow} \Omega; \mathcal{O} = \mathcal{H}; \hat{\phi} \overset{r}{\leftarrow} [\![\phi]\!]_{\mathcal{H},\mathcal{A}} : \hat{\phi}\right]$

When $pvar(\phi) = \emptyset$, semantics of ϕ does not depend on the adversary $\mathcal{A}$ and we will use the notation $[\![\phi]\!]$ (or $[\![\phi]\!]_{\mathcal{H}}$) instead of $[\![\phi]\!]_{\mathcal{A}}$ (respectively $[\![\phi]\!]_{\mathcal{H},\mathcal{A}}$).

3.3 Soundness and Completeness

The computational model of a cryptographic scheme is closer to reality than its formal representation by being a more detailed description. Therefore, the accuracy of a formal model can be characterized based on how close it is to the computational model. For this reason, we introduce the notions of soundness and completeness (inspired from [8]) that relate relations in the symbolic model with respect to similar relations in the computational model. Let E be an equivalence theory and let $R_1 \subseteq \mathbf{T} \times \mathbf{T}$, $R_2 \subseteq \mathbf{F} \times \mathbf{T}$, and $R_3 \subseteq \mathbf{F} \times \mathbf{F}$ be relations on closed frames, on closed terms, and relations on closed frames and terms, respectively.

- R_1 is $=$-sound iff for all terms T_1, T_2 of the same sort, $(T_1, T_2) \in R_1$ implies that $\mathsf{Pr}[\hat{e_1}, \hat{e_2} \overset{r}{\leftarrow} [\![T_1, T_2]\!]_{\mathcal{A}} : \hat{e}_1 \neq \hat{e}_2))]$ is negligible for any ppt adversary $\mathcal{A}$.

- R_1 is $=$-complete iff for all terms T_1, T_2 of the same sort, $(T_1, T_2) \notin R_1$ implies that $\mathsf{Pr}[\hat{e_1}, \hat{e_2} \overset{r}{\leftarrow} [\![T_1, T_2]\!]_{\mathcal{A}} : \hat{e}_1 \neq \hat{e}_2))]$ is non-negligible for some ppt adversary $\mathcal{A}$.

- R_1 is $=$-faithful iff for all terms T_1, T_2 of the same sort, $(T_1, T_2) \notin R_1$ implies that $\mathsf{Pr}[\hat{e_1}, \hat{e_2} \overset{r}{\leftarrow} [\![T_1, T_2]\!]_{\mathcal{A}} : \hat{e}_1 = \hat{e}_2))]$ is negligible for any ppt adversary $\mathcal{A}$.

- R_2 is $\not\vdash$-sound iff all frame ϕ and term T, $(\phi, T) \in R_2$ implies that $\mathsf{Pr}[\hat{\phi}, \hat{e} \overset{r}{\leftarrow} [\![\phi, T]\!]_{\mathcal{A}} : \mathcal{A}^{\mathcal{O}}(\hat{\phi}) = \hat{e}]$ is negligible for any ppt adversary $\mathcal{A}$.

- R_2 is $\not\vdash$-complete iff for all frame ϕ and term T, $(\phi, T) \notin R_2$ implies that $\mathsf{Pr}[\hat{\phi}, \hat{e} \overset{r}{\leftarrow} [\![\phi, T]\!]_{\mathcal{A}} : \mathcal{A}^{\mathcal{O}}(\hat{\phi}) = \hat{e}]$ is non-negligible for some ppt adversary $\mathcal{A}$.

- R_3 is $\approx_E$-sound iff for all frames ϕ_1, ϕ_2 with the same domain, $(\phi_1, \phi_2) \in R_3$ implies that $([\![\phi_1]\!]_{\mathcal{A}}) \sim ([\![\phi_2]\!]_{\mathcal{A}})$ for any ppt adversary $\mathcal{A}$.

- R_3 is $\approx_E$-complete iff for all frames ϕ_1, ϕ_2 with the same domain, $(\phi_1, \phi_2) \notin R_3$ implies that $([\![\phi_1]\!]_{\mathcal{A}}) \not\approx ([\![\phi_2]\!]_{\mathcal{A}})$ for some ppt adversary $\mathcal{A}$.

4 Formal relations

One challenge of the paper is to propose appropriate symbolic relations that correctly abstract computational properties as indistinguishability of two distributions or weak secrecy of some random value (the adversary has only negligible probability to compute it). In this section we provide two symbolic relations (called formal indistinguishability relation and formal non-derivability relation) that are sound abstractions for the two above computational properties.

First we define well-formed relations and we recall a simplified definition of a formal indistinguishability relation as proposed in [7].

Definition 5 (Well-formed relations). *A relation* $S_d \subseteq \boldsymbol{F} \times \boldsymbol{T}$ *is called* **well-formed** *if* $fn(M) \subseteq n(\phi)$ *for any* $(\phi, M) \in S_d$*, and a relation* $S_i \subseteq \boldsymbol{F} \times \boldsymbol{F}$ *is* **well-formed** *if* $dom(\phi_1) = dom(\phi_2)$ *for any* $(\phi_1, \phi_2) \in S_i$*.*

Definition 6. *[FIR [7]] A well-formed relation* $\cong \subseteq \boldsymbol{F} \times \boldsymbol{F}$ *is called a* **formal indistinguishability relation (FIR for short)** *with respect to the equational theory* $=_E$*, if* $\cong$ *is closed with respect to the following closure rules:*
(GE1) If $\phi_1 \cong \phi_2$ *then* $\phi\phi_1 \cong \phi\phi_2$*, for any frame* ϕ *such that* $var(\phi) \subseteq dom(\phi_i)$ *and* $n(\phi) \cap n(\phi_i) = \emptyset$*.*
(GE2) $\phi \cong \phi'$ *for any frame* ϕ' *such that* $\phi' =_E \phi$*.*
(GE3) $\tau(\phi) \cong \phi$ *for any renaming* τ*.*

This definition is a good starting point to capture indistinguishability in the following sense: if we have a correct implementation of the abstract algebra (i.e. $=_E$ is $=$-sound) and we were provided with some initial relation S (reflecting some computational assumption) which is $\approx$-sound , then the closure of S using the above rules produces a larger relation which still remains $\approx$-sound. But in order to use this definition for real cryptographic constructions , we need to enrich it in several aspects. First, most of constructions which are proposed in the literature, ([9], [28], [22], [24], [26], [10]) use bijective functions (XOR-function or permutations) as basic bricks. To deal with these constructions, we add the following closure rule:
(GE4) If M, N are terms such that $N[M/z] =_E y$, $M[N/y] =_E z$, $var(M) = \{y\}$ and $var(N) = \{z\}$, then for any substitution σ such that $r \notin (fn(\sigma) \cup fn(M) \cup fn(N))$ and $x \notin dom(\sigma)$ it holds $\nu\widetilde{n}.r.\{\sigma, x = M[r/y]\} \cong \nu\widetilde{n}.r.\{\sigma, x = r\}$.

Second, cryptographic constructions use often hash functions. In ideal models, if one applies a hash function (modeled by random functions [10] or pseudo-random permutations [23]) to a argument that is weakly secret, it returns a random value. And they are quite frequent primitives in cryptography that only ensure weak secrecy. One-way functions only guarantee that an adversary that possesses the image by a one-way function of a random value, has only a negligible probability to compute this value. The computational Diffie-Hellman (CDH)

assumption states that if given the tuple g, g^a, g^b for some randomly-chosen generator g and some random values a, b, it is computationally intractable to compute g^{a*b} (equivalently g^{a*b} is a weakly secret value). This motivates us to introduce the **formal non-derivability relation** as an abstraction of weak secrecy. Let us explain the basic closure rules of this relation. Since we assume that all sorts are implemented by large finite sets of bit strings, it is clearly that *(GD1)* $\nu r.\emptyset \not\vdash r$.

Renaming does not change the concrete semantics of terms or frames.
(GD2) If $\phi \not\vdash M$ then $\tau(\phi) \not\vdash \tau(M)$ for any renaming τ.

If the equational theory is preserved in the computational world, then equivalent terms or frames are indistinguishable.
(GD3) If $\phi \not\vdash M$ then $\phi \not\vdash N$ for any term $N =_E M$.
(GD4) If $\phi \not\vdash M$ then $\phi' \not\vdash M$ for any frame $\phi' =_E \phi$.

If some bit string (concrete implementation of term M) is weakly secret, then any polynomially computation (abstracted by the frame ϕ') does not change this.
(GD5) If $\phi \not\vdash M$ then $\phi'\phi \not\vdash M$ for any frame ϕ' such that $n(\phi') \cap n(\phi) = \emptyset$.

Next rule gives a relationship between indistiguishability and secrecy: if two distributions are indistinguishable, then they leak exactly the same information.
(GD6) For all substitutions σ_1, σ_2 such that $x \notin dom(\sigma_i)$, if $\nu\widetilde{n}.\{\sigma_1, x = M\} \cong \nu\widetilde{n}.\{\sigma_2, x = N\}$ and $\nu\widetilde{n}.\sigma_1 \not\vdash M$ then $\nu\widetilde{n}.\sigma_2 \not\vdash N$.

If the concrete implementation of the symbolic contextual term $T(z)$ is a feasible computation, that is, the adversary has all the needed information to compute $T(\cdot)$ $(fn(T) \cap n(\phi) = \emptyset)$, then the concrete implementation of $(T\phi)[M/z]$ is weakly secret only because the implementation of M itself is weakly secret.
(GD7) If $\phi \not\vdash (T\phi)[M/z]$ then $\phi \not\vdash M$, where T is such that $fn(T) \cap n(\phi) = \emptyset$.

One can remark now that *(GD6)* may be generalized to the rule below
(GD6g) If T, U are terms such that $(fn(T) \cup fn(U)) \cap \widetilde{n} = \emptyset$, $z \in var(T) \setminus var(U)$ and $U[T/y] =_E z$, then for all substitutions σ_1, σ_2 such that $x \notin dom(\sigma_i)$ and $\nu\widetilde{n}.\{\sigma_1, x = T[M/z]\} \cong \nu\widetilde{n}.\{\sigma_2, x = T[N/z]\}$ and $\nu\widetilde{n}.\sigma_1 \not\vdash M$ then $\nu\widetilde{n}.\sigma_2 \not\vdash N$.

Actually, *(GD6g)* is consequence of rules *(GD3)*, *(GD6)* and *(GD7)*.

Now the rules that capture hash functions in the ROM: the image by a random function of a weakly secret value is a completely random value.
(HD1) If $\nu\widetilde{n}.r.\sigma[r/h(T)] \not\vdash T$ and $r \notin n(\sigma)$, and if $\sigma[r/h(T)]$ does not contain any subterm of the form $h(\bullet)$, then $\nu\widetilde{n}.\sigma \not\vdash T$.
(HE1) If $\nu\widetilde{n}.r.\sigma[r/h(T)] \not\vdash T$ and $r \notin n(\sigma)$, and if $\sigma[r/h(T)]$ does not contain any subterm of the form $h(\bullet)$, then $\nu\widetilde{n}.r.\sigma \cong \nu\widetilde{n}.r.\sigma[r/h(T)]$.

The definition below formalizes the tight connection between FIR and FNDR.

Definition 7 (FNDR and FIR). *A pair of well formed relations* $(\not\vdash, \cong)$ *is a pair of* **(formal non-derivability relation, formal indistinguishability relation)** *with respect to the equational theory* $=_E$*, if* $(\not\vdash, \cong)$ *is closed with respect to the rules (GD1), ..., (GD7),(GE1),...,(GE4), (HD1),(HE1) and* $\cong$ *is an equivalence.*

The theorem 1 shows that if a pair (FIR,FNDR) was generated by relations S_d and S_i, then it is sufficient to check only soundness of elements in S_d and S_i to ensure that the closures $\langle S_d \rangle_{\not\vdash}$ and $\langle S_i \rangle_{\cong}$ are sound. We define $(D_1, I_1) \sqsubset (D_2, I_2)$ if and only if $D_1 \subseteq D_2$ and $I_1 \subseteq I_2$. It is easy to see that $\sqsubset$ is an order.

Theorem 1. *Let (S_d, S_i) be a well-formed pair of relations. Then, it exists a unique smallest (with respect to $\sqsubset$) pair denoted $(\langle S_d \rangle_{\not\vdash}, \langle S_i \rangle_{\cong})$ of (FNDR, FIR) such that $\langle S_d \rangle_{\not\vdash} \supseteq S_d$ and $\langle S_i \rangle_{\cong} \supseteq S_i$. In addition, if $=_E$ is $=$-sound, S_d is $\not\vdash$-sound and S_i is $\approx$-sound, then also $\langle S_d \rangle_{\not\vdash}$ is $\not\vdash$-sound and $\langle S_i \rangle_{\cong}$ is $\approx$-sound.*

The reader should notice that rules *(HE1)* and *(HD1)* can be strengthened if $=_E$ is $=$-faithful: "if $\sigma[r/h(T)]$ does not contain any subterm of the form $h(\bullet)$" can be replaced with "$T \neq_E T'$ for any subterm $h(T')$ of $\sigma[r/h(T)]$".

5 Applications

We apply the framework of Section 4 in order to prove IND-CPA security of several generic constructions for asymmetric encryptions. So we will consider pairs of relations $(\not\vdash, \cong) = (\langle S_d \rangle_{\not\vdash}, \langle S_i \rangle_{\cong})$ generated by some initial sets (S_d, S_i), in different equational theories. We assume that all $=_E$, S_d, S_i that are considered in this section satisfy the conditions of Theorem 1. We emphasize the following fact: adding other equations than those considered does not break the computational soundness of results proved in this section, as long as the computational hypothesis encoded by S_d and S_i still hold.
First we introduce a general abstract algebra that we will extend in order to cover different constructions. We consider three sorts $Data$, $Data^1$, $Data^2$, and the symbols $|| : Data^1 \times Data^2 \to Data$, $\oplus_S : S \times S \to S$, $0_S : S$, with $S \in \{Data, Data^1, Data^2\}$ and $\pi_j : Data \to Data^j$, with $j \in \{1, 2\}$. For simplicity, we omit S when using $\oplus_S$ or 0_S . The equational theory E_g is generated by:
(XEq1) $x \oplus 0 =_{E_g} x$ *(XEq2)* $x \oplus y =_{E_g} y \oplus x$ *(PEq1)* $\pi_1(x||y) =_{E_g} x$
(XEq2) $x \oplus x =_{E_g} 0$ *(XEq4)* $x \oplus (y \oplus z) =_{E_g} (x \oplus y) \oplus z$ *(PEq2)* $\pi_2(x||y) =_{E_g} y$

$||$ is intended to model concatenation, $\oplus$ is the classical XOR and π_j are the projections. Next rules are consequences of the closure rules from Section 4.
(SyE) If $\phi_1 \cong \phi_2$ then $\phi_2 \cong \phi_1$.
(TrE) If $\phi_1 \cong \phi_2$ and $\phi_2 \cong \phi_3$ then $\phi_1 \cong \phi_3$.
(XE1) If $r \notin (fn(\sigma) \cup fn(T))$ then $\nu\widetilde{n}.r.\{\sigma, x = r \oplus T\} \cong \nu\widetilde{n}.r.\{\sigma, x = r\}$.
(CD1) If $(\phi \not\vdash T_1 \vee \phi \not\vdash T_2)$ then $\phi \not\vdash T_1||T_2$.
(XD1) If $\nu\widetilde{n}.\sigma \not\vdash T$ and $r \notin (\widetilde{n} \cup fn(T))$ then $\nu\widetilde{n}.r.\{\sigma, x = r \oplus T\} \not\vdash T$.

5.1 Trapdoor one-way functions in the symbolic model

We extend the above algebra in order to model trapdoor one-way functions. We add a sort $iData$ and new symbols $f : Data \times Data \to iData$,$f^{-1} : iData \times Data \to Data, pub : Data \to Data$. f is a trapdoor permutation, with f^{-1} being the inverse function. We extend the equational theory:
(OEq1) $f^{-1}(f(x, pub(y)), y) =_{E_g} x$.

To simplify the notations, we will use $f_k(\bullet)$ instead of $f(\bullet, pub(k))$. Now we want to capture the one wayness of function f. Computationally, a one-way function only ensures the weakly secrecy of a random argument r (as long as the key k is not disclosed to the adversary). Hence we define $S_i = \emptyset$ and $S_d = \{(\nu k.r.\{x_k = pub(k), x = f_k(r)\}, r)\}$.

The following frame encodes the Bellare-Rogaway encryption scheme ([10]):
$\phi_{br}(m) = \nu k.r.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus m, z = h(m||r)\}$
where m is the plaintext to be encrypted, f is a trapdoor one-way function, and g and h are hash functions (hence oracles in the ROM model).

Now we can see the necessity of p-variables in order to encode IND-CPA security of an encryption scheme. It is not enough to prove that for any two messages m_1 and m_2 the following equivalence holds:

$$\nu k.r.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus m_1, z = h(m_1||r)\} \cong$$
$$\nu k.r.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus m_2, z = h(m_2||r)\}$$

We did not capture that the adversary is adaptive and she can choose her challenges depending on the public key. We must prove a stronger equivalence: for any terms $p(x_k)$ and $p'(x_k)$,

$$\nu k.r.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus p(x_k), z = h(p(x_k)||r)\} \cong$$
$$\nu k.r.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus p'(x_k), z = h(p'(x_k)||r)\}$$

The reader noticed that for asymmetric encryption, this suffices to ensure IND-CPA: possessing the public key and having access to hash-oracles allow to encrypt any message (having an oracle to encrypt messages becomes superfluous).

Actually, it suffices to prove $\nu k.r.s.t.\{x_k = pub(k), x_a = f_k(r), y = g(r) \oplus p(x_k), z = h(p(x_k)||r)\} \cong \nu k.r.s.t.\{x_k = pub(k), x_a = f_k(r), y = s, z = t\}$. By transitivity, this implies: for any two challenges that adversary chooses for $p(x_k)$, the distributions she gets are indistinguishable.

Next rules are consequences of the definition of S_d and of the closure rules.
(OD1) If f is a one-way function, then $\nu k.r.\{x_k = pub(k), x = f_k(r)\} \not\vdash r$.
(ODg1) If f is a one-way function and $\nu\widetilde{n}.\nu k.\{x_k = pub(k), x = T\} \cong \nu r.\nu k.\{x_k = pub(k), x = r\}$, then $\nu\widetilde{n}.\nu k.\{x_k = pub(k), x = f_k(T)\} \not\vdash T$.

The proof of IND-CPA security of Bellare-Rogaway scheme is presented in Figure 1. To simplify the notations, implicitly, all names in frames are restricted and we note $\sigma_2 \equiv x_k = pub(k), x_a = f_k(r)$, and $\sigma_3 \equiv \sigma_2, y = g(r) \oplus p(x_k)$.

5.2 Partially one-way functions in the symbolic model

In this subsection, we show how we can deal with trapdoor partially one-way functions ([24]). We demand for function f a stronger property than one-wayness. Let $Data_1$ be a new sort, and let $f : Data_1 \times Data \times Data \to iData$ and $f^{-1} : iData \times Data \to Data_1$ be functions such that
(OEq1) $f(f^{-1}(x, y), z, pub(y)) =_{E_g} x$.

The function f is said *partially one way*, if for any given $f(r, s, pub(k))$, it is impossible to compute in polynomial time a corresponding r without the trapdoor k. In order to deal with fact that f is partially one-way, we define $S_i = \emptyset$ and $S_d = \{(\nu k.r.s.\{x_k = pub(k), x = f_k(r, s)\}, r)\}$.
The frame below encodes the encryption scheme proposed by Pointcheval ([24]).
$\phi_{po}(m) = \nu k.r.s.\{x_k = pub(k), x_a = f_k(r, h(m||s)), y = g(r) \oplus (m||s)\}$
where m is the plaintext to be encrypted, f is a trapdoor partially one-way function, and g and h are hash functions. To prove IND-CPA security of this

$$\textit{TrE}\;\dfrac{\textit{HE1}\;\dfrac{\textit{CD1}\;\dfrac{\textit{GD5}\;\dfrac{\textit{HD1}\;\dfrac{\textit{GD5}\;\dfrac{\textit{OD1}\;\dfrac{}{\{\sigma_2\} \not\vdash r}}{\{\sigma_2, y = s'\} \not\vdash r}}{\{\sigma_2, y = g(r)\} \not\vdash r}}{\{\sigma_2, y = g(r) \oplus p(x_k), z = t\} \not\vdash r}}{\{\sigma_2, y = g(r) \oplus p(x_k), z = t\} \not\vdash p(x_k)||r}}{\{\sigma_2, y = g(r) \oplus p(x_k), z = h(p(x_k)||r)\} \cong \{\sigma_2, y = g(r) \oplus p(x_k), z = t\}} \qquad (T1)}{\{\sigma_2, y = g(r) \oplus p(x_k), z = h(p(x_k)||r)\} \cong \{x_k = pub(k), x_a = f_k(r), y = s, z = t\}}$$

Fig. 1. Proof of IND-CPA security of Bellare-Rogaway scheme.

$$\textit{GE1}\;\dfrac{\textit{TrE}\;\dfrac{\textit{GE1}\;\dfrac{\textit{HE1}\;\dfrac{\textit{GD5}\;\dfrac{\textit{OD1}\;\dfrac{}{\{\sigma_2\} \not\vdash r}}{\{\sigma_2, y = s\} \not\vdash r}}{\{\sigma_2, y = g(r)\} \cong \{\sigma_2, y = s\}}}{\{\sigma_3\} \cong \{\sigma_2, y = s \oplus p(x_k)\}} \quad \textit{XE1}\;\dfrac{}{\{\sigma_2, y = s \oplus p(x_k)\} \cong \{\sigma_2, y = s\}}}{\{\sigma_2, y = g(r) \oplus p(x_k)\} \cong \{\sigma_2, y = s\}}}{\{\sigma_2, y = g(r) \oplus p(x_k), z = t\} \cong \{\sigma_2, y = s, z = t\}}$$

Fig. 2. Tree $(T1)$ from Figure 1.

scheme, we show that $\nu k.r.s.s_1.s_2\{x_k = pub(k), x_a = f_k(r, h(p(x_k)||s)), y = g(r) \oplus (p(x_k)||s)\} \cong \nu k.r.s.s_1.s_2.\{x_k = pub(k), x_a = f_k(r, s_1), y = s_2\}$.

Next rule is a consequence of the definition of S_d.

(ODp1) If f is a one-way function, then $\nu k.r.s.\{x_k = pub(k), x = f_k(r, s)\} \not\vdash r$. The proof of IND-CPA security of Pointcheval scheme is presented in Figure 3. To simplify notations we suppose that all names in frames are restricted and we note $\sigma_2 \equiv x_k = pub(k), x_a = f_k(r, h(p(x_k)||s))$ and $\sigma_3 \equiv \sigma_2, y = s_2 \oplus (p(x_k)||s)$.

5.3 Computational Diffie Hellman (CDH) Assumption

In this subsection we prove IND-CPA security of a variant of Hash-ElGamal encryption scheme ([27]) in the random oracle model under the CDH assumption. The proof of the original scheme([6]) can be easily obtained from our proof and it can be done entirely in our framework. We will consider two sorts G and A, symbol functions $exp : G \times A \to G$, $* : A \times A \to A$, $0_A : A$, $1_A : A$, $1_G : G$. We write M^N instead of $exp(M, N)$. We extend E_g by the following equations:
(XEqe1) $(x^y)^z =_{E_g} x^{y*z}$. *(XEqe2)* $x^{1_A} =_{E_g} x$. *(XEqe3)* $x^{0_A} =_{E_g} 1_G$.
To capture the CDH Assumption in the symbolic model we define $S_i = \emptyset$ and $S_d = \{(\nu g.r.s.\{x_g = g, x = g^s, y = g^r\}, g^{s*r})\}$. Then we get the next rule:
(CDH) $\nu g.r.s.\{x_g = g, x = g^s, y = g^r\} \not\vdash g^{s*r}$.

The following frame encodes the Hash-ElGamal encryption scheme.
$\phi_{hel}(m) = \nu g.r.s.\{x_g = g, x = g^s, y = g^r, z = h(g^{s*r}) \oplus m\}$
where m is the plaintext to be encrypted, (g, g^s) is the public key and h is a hash function. The proof of IND-CPA security of Hash-ElGamal's scheme is provided in Figure 6. We supposed that all names are restricted and we noted $\sigma_e \equiv x_g = g, x = g^s, y = g^r$, and $\sigma_f \equiv \sigma_e, z = t \oplus p(x, x_g)$.

$$\mathit{TrE}\ \dfrac{(T2) \qquad\qquad (T3)}{\{\sigma_2, y = g(r) \oplus (p(x_k)||s)\} \cong \{x_k = pub(k), x_a = f_k(r, s_1), y = s_2\}}$$

Fig. 3. Proof of IND-CPA security of Pointcheval scheme.

$$\mathit{HE1}\ \dfrac{\mathit{GD6}\ \dfrac{\mathit{SyE}\ \dfrac{\mathit{XE1}\ \dfrac{}{\{\sigma_3, x = r\} \cong \{\sigma_2, y = s_2, x = r\}}}{\{\sigma_2, y = s_2, x = r\} \cong \{\sigma_3, x = r\}} \quad \mathit{GD5}\ \dfrac{\mathit{ODp1}\ \dfrac{}{\{\sigma_2\} \nvdash r}}{\{\sigma_2, y = s_2\} \nvdash r}}{\{\sigma_3\} \nvdash r}}{\{\sigma_2, y = g(r) \oplus (p(x_k)||s)\} \cong \{\sigma_3\}}$$

Fig. 4. Tree $(T2)$ from Figure 3.

$$\mathit{TrE}\ \dfrac{\mathit{XE1}\ \dfrac{}{\{\sigma_3\} \cong \{\sigma_2, y = s_2\}} \quad \mathit{GE1}\ \dfrac{\mathit{HE1}\ \dfrac{\mathit{CD1}\ \dfrac{\mathit{GD5}\ \dfrac{\mathit{GD1}\ \dfrac{}{\emptyset \nvdash s}}{\{x_k = pub(k), x_a = f_k(r, s_1)\} \nvdash s}}{\{x_k = pub(k), x_a = f_k(r, s_1)\} \nvdash p(x_k)||s}}{\{\sigma_2\} \cong \{x_k = pub(k), x_a = f_k(r, s_1)\}}}{\{\sigma_2, y = s_2\} \cong \{x_k = pub(k), x_a = f_k(r, s_1), y = s_2\}}}{\{\sigma_3\} \cong \{x_k = pub(k), x_a = f_k(r, s_1), y = s_2\}}$$

Fig. 5. Tree $(T3)$ from Figure 3.

$$\mathit{TrE}\ \dfrac{\mathit{GE1}\ \dfrac{\mathit{HE1}\ \dfrac{\mathit{GD5}\ \dfrac{\mathit{CDH}\ \dfrac{}{\{\sigma_e\} \nvdash g^{s*r}}}{\{\sigma_e, z = t\} \nvdash g^{s*r}}}{\{\sigma_e, z = h(g^{s*r})\} \cong \{\sigma_e, z = t\}}}{\{\sigma_e, z = h(g^{s*r}) \oplus p(x, x_g)\} \cong \{\sigma_f\}} \quad \mathit{XE1}\ \dfrac{}{\{\sigma_f\} \cong \{\sigma_e, z = t\}}}{\{x_g = g, x = g^s, y = g^r, z = h(g^{s*r}) \oplus p(x, x_g)\} \cong \{x_g = g, x = g^s, y = g^r, z = t\}}$$

Fig. 6. Proof of IND-CPA security of Hash-ElGamal's scheme

6 Static equivalence and FIR

In this section we adapt the definition of deductibility and static equivalence ([8]) to our framework. After, we justify why they are too coarse to be appropriate abstractions for indistinguishability and weak secrecy. Actually, Proposition 1 states that they are coarser approximations of indistinguishability and weak secrecy than FIR and FNDR.

If ϕ is a frame, and M, N are terms, then we use $(M =_E N)\phi$ for $M\phi =_E N\phi$.

Definition 8 (Deductibility). *A (closed) term T is* **deductible** *from a frame ϕ where $(p_i)_{i \in I} = pvar(\phi)$, written $\phi \vdash T$, if and only if there exists a term M and a set of terms $(M_i)_{i \in I}$, such that $var(M) \subseteq dom(\phi)$, $ar(M_i) = ar(p_i)$, $fn(M, M_i) \cap n(\phi) = \emptyset$ and $(M =_E T)(\phi[(M_i(T_{i_1}, \ldots, T_{i_k})/p_i(T_{i_1}, \ldots, T_{i_k}))_{i \in I}])$. We denote by $\nvdash$ the logical negation of $\vdash$.*

For instance, we consider the frame $\phi = \nu k_1.k_2.s_1.s_2.\{x_1 = k_1, x_2 = k_2, x_3 = h((s_1 \oplus k_1) \oplus p(x_1, x_2)), x_4 = h((s_2 \oplus k_2) \oplus p(x_1, x_2))\}$ and the equational theory E_g. Then $h(s_1) \oplus k_2$ is deductible from ϕ since $h(s_1) \oplus k_2 =_{E_g} x_3[x_1/p(x_1, x_2)] \oplus x_2$ but $h(s_1) \oplus h(s_2)$ is not deductible.

If we consider the frame $\phi' = \nu k.r.s.\{x_k = pub(k), x = f_k(r||s)\}$ where f is a trapdoor one-way function, then neither $r||s$, nor r is deductible from ϕ'. The one-wayness of f is modelled by the impossibility of inverting f if k is not disclosed. While this is fair for $r||s$ according to the computational guarantees of f, it seems too strong of assuming that r alone cannot be computed if f is "just" one-way. This raises some doubts about the fairness of $\not\vdash$ as a good abstraction of weak secrecy. We can try to correct this and add an equation of the form $g(f(x||z, pub(y)), y) =_{E_g} x$. And now, what about r_1, if one gives $f((r_1||r_2)||s)$? In the symbolic setting r_1 is not deductible; computationally, we have no guarantee; hence, when one stops to add equations? Moreover, in this way we could exclude "good" one-way functions: computationally, if f is a one-way function, then $f'(x||y) \overset{def}{=} x||f(y)$, is another one-way function. The advantage of defining non-deductibility as we did it in the Section 4, is that first, we capture "just" what is supposed to be true in the computational setting, and second, if we add more equations to our abstract algebra (because we discovered that the implementation satisfies more equations) in a coherent manner with respect to the initial computational assumptions, then our proofs still remain computationally sound. This is not true for $\not\vdash$.

Definition 9. *A* **test** *for a frame ϕ is a tuple $\Upsilon = ((M_i)_{i\in I}, M, N)$ such that $ar(M_i) = ar(p_i)$, $var(M, N) \subseteq dom(\phi)$, $fn(M, N, M_i) \cap n(\phi) = \emptyset$. Then ϕ* **passes** *Υ if and only if $(M =_E N)(\phi[(M_i(T_{i_1}, \ldots, T_{i_k})/p_i(T_{i_1}, \ldots, T_{i_k}))_{i\in I}])$.*

Definition 10 (Statically Equivalent). *Two frames ϕ_1 and ϕ_2 are* **statically equivalent**, *written as $\phi_1 \approx_E \phi_2$, if and only if*
(i) $dom(\sigma_1) = dom(\sigma_2)$;
(ii) for any test Υ, ϕ_1 passes the test Υ if and only if ϕ_2 passes the test Υ.

For instance, the two frames $\phi_1 = \nu k.s.\{x_1 = k, x_2 = h(s) \oplus (k \oplus p(x_1))\}$ and $\phi_2 = \nu k.s.\{x_1 = k, x_2 = s \oplus (k \oplus p(x_1))\}$ are statically equivalent with respect to E_g. However the two frames $\phi'_1 = \nu k.s.\{x_1 = k, x_2 = h(s) \oplus (k \oplus p(x_1)), x_3 = h(s)\}$ and $\phi'_2 = \nu k.s.\{x_1 = k, x_2 = s \oplus (k \oplus p(x_1)), x_3 = h(s)\}$ are not. The frame ϕ'_2 passes the test $((x_1), x_2, x_3)$, but ϕ'_1 does not.

Let us now consider the equational theory from subsection 5.2. Then the following frames $\nu g.a.b.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^{a*b})$ and $\nu g.a.b.c.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^c)$ are statically equivalent. This seems right, it is the DDH assumption: a computational implementation that satisfies indistinguishability for the interpretations of this two frames will simply satisfy the DDH assumption. But soundness would imply much more. Even $\nu g.a.b.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^{a^2*b^2}\}$ and $\nu g.a.b.c.\{x_1 = g, x_2 = g^a, x_3 = g^b, x_4 = g^c\}$ will be statically equivalent. It is unreasonable to assume that this is true for the computational setting. As for non-deductibility, the advantage of considering FIR as the abstraction of indistinguishability, is that if we add equations in a coherent manner with respect to the initial computational assumptions (that is with S_i), then our proofs still remain computationally sound. The proposition below says that if we consider initial reasonable sets S_d and S_i, then we get finer approximations of indistinguishability and weak secrecy than $\not\vdash$ and $\approx_E$.

Proposition 1. *Let (S_d, S_i) be such that $S_d \subseteq\not\vdash$ and $S_i \subseteq\approx_E$. Then $\langle S_d\rangle_{\not\vdash} \subseteq\not\vdash$ and $\langle S_i\rangle_{\cong} \subseteq\approx_E$.*

7 Conclusion

In this paper we developed a general framework for relating formal and computational models for generic encryption schemes in the random oracle model. We proposed general definitions of formal indistinguishability relation and formal non-derivability relation, that is symbolic relations that are computationally sound by construction. We extended previous work with respect to several aspects. First, our framework can cope with adaptive adversaries. This is mandatory in order to prove IND-CPA security. Second, many general constructions use one-way functions, and often they are analyzed in the random oracle model: hence the necessity to capture the weak secrecy in the computational world. Third, the closure rules we propose are designed with the objective of minimizing the initial relations which depend of the cryptographic primitives and assumptions. We illustrated our framework on several generic encryption schemes: we proved IND-CPA security of the scheme proposed by Bellare and Rogaway in [10], of Hash El Gamal [6] and of the scheme proposed by Pointcheval in [24].

As future works, we project to study the (relative) completeness of various equational symbolic theories. Other extensions will be to capture fully active adversaries or exact security (as in [11], we could define indistinguishabiliy as up-to some explicit probability p instead of up-to a negligible probability).

References

1. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS)*, Sendai, Japan, 2000. Springer-Verlag.
2. Martín Abadi, Mathieu Baudet, and Bogdan Warinschi. Guessing attacks and the computational soundness of static equivalence. In *FoSSaCS*, volume 3921 of *LNCS*, pages 398–412. Springer, 2006.
3. Martín Abadi and Andrew D. Gordon. A bisimulation method for cryptographic protocols. In *ESOP*, volume 1381 of *LNCS*, pages 12–26. Springer, 1998.
4. M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable dolev-yao style cryptographic library. In *CSFW*, pages 204–218. IEEE , 2004.
5. M. Backes, B. Pfitzmann, and M. Waidner. Symmetric authentication within a simulatable cryptographic library. In *ESORICS*, volume 2808 of *LNCS*, pages 271–290. Springer, 2003.
6. Joonsang Baek, Byoungcheon Lee, and Kwangjo Kim. Secure length-saving elgamal encryption under the computational diffie-hellman assumption. In *ACISP*, volume 1841 of *LNCS*, pages 49–58. Springer, 2000.
7. Gergei Bana, Payman Mohassel, and Till Stegers. Computational soundness of formal indistinguishability and static equivalence. In Mitsu Okada and Ichiro Satoh, editors, *ASIAN*, volume 4435 of *LNCS*, pages 182–196. Springer, 2006.

8. Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. In *ICALP*, volume 3580 of *LNCS*, pages 652–663. Springer, 2005.
9. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT'04*, volume 950 of *LNCS*, pages 92–111, 1994.
10. Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS'93*, pages 62–73, 1993.
11. Bruno Blanchet and David Pointcheval. Automated security proofs with sequences of games. In *CRYPTO'06*, volume 4117 of *LNCS*, pages 537–554, 2006.
12. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.
13. Ran Canetti and Jonathan Herzog. Universally composable symbolic analysis of mutual authentication and key-exchange protocols. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *LNCS*, pages 380–403. Springer, 2006.
14. V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In Sagiv [25], pages 157–171.
15. Judicaël Courant, Marion Daubignard, Cristian Ene, Pascal Lafourcade, and Yassine Lakhnech. Towards automated proofs for asymmetric encryption schemes in the random oracle model. In *CCS'2008*, pages 371–380. ACM, 2008.
16. D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
17. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *J. Cryptol.*, 1(2):77–94, 1988.
18. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.
19. R. Janvier, Y. Lakhnech, and L. Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In Sagiv [25], 172–185.
20. P. Laud. Symmetric encryption in automatic analyses for confidentiality against adaptive adversaries. In *Symposium on Security and Privacy*, pages 71–85, 2004.
21. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. *Theory of Cryptography Conference*, 133–151. Springer, 2004.
22. T. Okamoto and D. Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA'01*, pages 159–175, 2001.
23. Duong Hieu Phan and David Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In *Selected Areas in Cryptography*, volume 3357 of *LNCS*, pages 182–197. Springer, 2004.
24. D. Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In *PKC'00*, pages 129–146, 2000.
25. Shmuel Sagiv, editor. *Programming Languages and Systems, 14th European Symposium on Programming,ESOP 2005, April 4-8*, volume 3444 of *LNCS*, 2005.
26. V. Shoup. Oaep reconsidered. *J. Cryptology*, 15(4):223–249, 2002.
27. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. cryptology eprint archive, report 2004/332, 2004.
28. Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *J. on Selected Areas in Communications*, 11(5):715–724, 1993.