Steve Kremer and Laurent Mazaré

Adaptive Soundness of Static Equivalence

Research Report LSV-07-09

February 2007

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

CENTRE NATIONAL DE LA RECHERCHE

Adaptive Soundness of Static Equivalence*

Steve Kremer Laurent Mazaré
LSV, ENS Cachan & CNRS & INRIA Futurs
{kremer|mazare}@lsv.ens-cachan.fr

Abstract

We define a framework to reason about sound implementations of equational theories in the presence of an adaptive adversary. In particular, we focus on soundess of static equivalence. We illustrate our framework on several equational theories: symmetric encryption, XOR, modular exponentiation and also joint theories of encryption and modular exponentiation as well as encryption and XOR. For the last two examples we use a proof technique that enables us to reuse proofs for the separate theories. Finally, we define a model for symbolic analysis of dynamic group key exchange protocols, and show its computational soundness.

1 Introduction

It is well-known that security protocols are extremely error-prone, even for simple protocols. This is mainly due to the fact that they are executed in a hostile environment, such as the Internet. The need for rigorous proofs of such protocols was recognized very early and two distinct approaches for proving security protocols correct have been developed during the last 20 years. The symbolic approach considers an abstract model, where messages and cryptographic primitives are modeled by a term algebra and the adversary manipulates terms according to a pre-defined set of rules, typically an inference system. The computational approach considers a more detailed execution model. Protocol messages are modeled as bitstrings and cryptographic primitives are polynomial-time algorithms. The adversary is an arbitrary probabilistic polynomial-time Turing machine and the security of a protocol is measured as the success probability of such an adversary.

A considerable advantage of the symbolic model is that proofs can be (at least partially) automated. Unfortunately, it is not clear whether the abstract symbolic model captures all possible attacks. While the computational model provides much stronger security guarantees, proofs are generally harder and difficult to automate. Therefore a recent trend tries to get the best of both worlds: an abstract model which provides strong computational guarantees. In their seminal paper, Abadi and Rogaway [4] have shown a first such *soundness result* in the presence of a passive attacker for a simple abstract algebra with symmetric encryption. Their result states that whenever two terms are symbolically indistinguishable, then the distributions resulting out of the implementation of the two terms are indistinguishable by a computational adversary. There have been many extensions of this work. We discuss some of them below.

Recently, Baudet *et al*. [10] presented a general framework for reasoning about the soundness of the implementation of an equational theory. Rather than considering a fixed set of cryptographic primitives, they allow a specification by the means of an equational theory. The formal indistinguishability relation they consider is static equivalence, a well-established security notion coming from cryptographic pi calculi [3] whose verification can often be automated [2, 12]. Studying soundness of equational theories is motivated by the numerous recent works on extending the classical Dolev-Yao result with equations: many protocols use indeed algebraic properties of cryptographic primitives (see [19] for a survey on such algebraic properties). Showing a soundness result for an equational theory proves that indeed "enough" equations have been considered in the symbolic model, with respect to a given implementation.

In this paper we consider the question of soundness of static equivalence in the presence of an *adaptive adversary*, rather than a purely passive one. This extends the work by Baudet *et al*. in a similar way as the work of Micciancio and Panjwani [28] extended the work of Abadi and Rogaway [4]. An adaptive adversary is allowed to choose the messages whose implementation he will be given. The choice of the messages can hence depend on previously observed distributions. This allows us to analyze protocols in which the adversary does not intercept or inject messages but is allowed to alter the execution flow of the protocol. The fact that the adversary cannot control the network is obviously a restriction but corresponds to the hypothesis of

*Work partly supported by the ARA SSIA Formacrypt.

authenticated channels which may be implemented over ordinary channels using for instance techniques such as those proposed by Katz and Yung in [27].

More precisely the contributions of our paper are as follows. We define the notion of adaptive soundness of static equivalence in a general framework. The definition is parameterized by the equational theory and the concrete algebra implementing the symbolic model. Intuitively, adaptive soundness is defined by the following cryptographic game. The adversary provides a sequence of pairs of symbolic messages (t_0^i, t_1^i) to an oracle which returns the implementation of either t_0^i or t_1^i, depending on a challenge bit. It is important to note that the adversary can construct the next pair of messages depending on the answer of the oracle. Adaptive soundness requires that whenever the sequences $t_0^1, \ldots, t_0^n$ and $t_1^1, \ldots, t_1^n$ are statically equivalent, the adversary cannot guess the challenge bit with a probability significantly different from $1/2$.

We show that adaptive soundness implies soundness of static equivalence as defined in [10], *i.e.*, in the presence of a passive adversary. The converse is not true which reflects the intuition that an adaptive adversary has strictly more power than a passive one. However, the stronger notion of *unconditional* soundness, which guarantees security in an information-theoretic sense, coincides for adaptive and passive adversaries. We also provide a proof technique based on a composition result: it allows us to reuse soundness results of two disjoint abstract signatures and conclude soundness of the joint signature. While the conditions under which such a combination works are of course restrictive they nevertheless match cases of practical interest. Moreover, Arnaud *et al*. [6] recently showed that decidability of static equivalence composes for disjoint equational theories.

We give adaptive soundness results for several theories. First, we consider the classical theory of symmetric encryption. We show that it is sufficient for the encryption scheme to respect the standard IND-CPA security notion to obtain adaptive soundness. This is similar to the main result in [28], adapted to our framework. We moreover show that a strictly weaker, non-adaptive variant of IND-CPA is sufficient for non-adaptive soundness. As a second illustration we consider the theory of exclusive or (XOR). This result follows directly from the unconditional soundness result for XOR shown in [10] and our previous result on unconditional soundness. The third theory we study is modular exponentiation in an Abelian group. We show that the *Decisional Diffie-Hellman* (DDH) assumption is sufficient and necessary for both adaptive and passive soundness of static equivalence (which hence coincide in this case). Finally, we use our combination technique to derive adaptive soundness for the joint theories of encryption and XOR, as well as encryption and modular exponentiation. We believe these are the first adaptive soundness results for modular exponentiation and XOR. Their importance is motivated by real-life protocols such as SSL/TLS that rely on Diffie-Hellman key exchange and thus use modular exponentiation.

Finally, to illustrate the usefulness of adaptive adversaries we define a symbolic model for the analysis of dynamic group key exchange (DKE) protocols. A DKE protocol is a suite of protocols which allows three actions: exchange of an initial key between a group of users, joining and leaving the group. A typical example of DKE is the AKE1 protocol [14]. In our symbolic model we assume static corruption and allow the adversary to schedule these subprotocol and decide which users initially exchange the key, join, respectively leave the group. We use our adaptive soundness result to show that this symbolic model is sound with respect to a corresponding computational model.

Related work. This paper is most obviously related to the works by Baudet *et al*. [10] on soundness of equational theories in the presence of a passive adversary and Micciancio and Panjwani [28] who introduced soundness against adaptive adversaries, considering only symmetric encryption. Our paper generalizes both of these works. Abadi *et al*. [1] also use the framework of [10] to show soundness of an equational theory useful for reasoning about offline guessing attacks modeled in terms of static equivalence. In [9], Bana *et al*. argue that the notion of static equivalence is too coarse and not sound for many interesting equational theories. As an example they show that the DDH assumption is not sufficient to imply soundness of static equivalence. They introduce a general notion of *formal indistinguishability relation*. In this paper we prefer to stick to static equivalence which has the advantage of being a well-established, tool-supported equivalence relation. We address the problems highlighted in [9] by proving soundness for a restricted set of *well-formed* frames. Restricting the set of frames to be considered is a classical technique. For instance Abadi and Rogaway forbid expressions containing key cycles. We provide similar restrictions to obtain soundness of static equivalence for modular exponentiation under the DDH assumption. In [25], Herzog also studies soundness of Diffie-Hellman based key agreement in the strand space model. He shows that any formal attack can be mapped to an algorithm breaking the (computational) Diffie-Hellman assumption. Hence, he studies the converse problem (which we would rather refer to as completeness).

There have also been works considering an active adversary. Backes *et al*. [8, 7] prove the soundness of an abstract *cryptographic library* including primitives for digital signatures, symmetric and asymmetric encryption. These results provide strong *universally composable* guarantees. In a more classical framework, closer to automated tools[1],

[1] One may note that recent results [30] provide computer-aided support

there have also been soundness results for active adversaries [29, 20, 26]. These results do however not provide universal composability. An exception is the result by Herzog and Canetti [17] who show soundness with universal composability of an automated tool, but only for a restricted class of protocols. Different kinds of approaches have been taken by Datta *et al*. [21] who provide a computational semantics to the *Protocol Composition Logic* and Blanchet [13] who develops a tool that aims at directly generating cryptographic proofs via sequences of games. At the moment we are not aware of any general results for equational theories in the active case. One may note that considering an active adversary is generally technically more involved although incomparable to an adaptive adversary. Considering a both active and adaptive adversary is a challenging problem and a topic of active research.

Outline of the paper. In the next section we introduce our symbolic and computational models. In Section 3, we define adaptive soundness of static equivalence, show results on the relationship with passive soundness and introduce our combination technique. In Section 4, we provide soundness results for several theories and combinations of them. In Section 5, we give a soundness result for the analysis of dynamic group protocols. Finally, we draw conclusions and give directions for future research.

2 Abstract and computational algebras

In this section we introduce our model, which is the same up to some minor changes as in [10].

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* $\mathcal{S} = \{s, s_1 \ldots\}$ and a set of *symbols* $\mathcal{F} = \{f, f_1 \ldots\}$ together with arities of the form $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$, $k \geq 0$. Symbols that take $k = 0$ arguments are called *constants*; their arity is simply written s. We fix a set of *names* $\mathcal{N} = \{a, b \ldots\}$ and a set of *variables* $\mathcal{X} = \{x, y \ldots\}$. We assume that names and variables are given with sorts, 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

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

for proofs in the framework of Backes *et al.*

where for the last case, we further require that t_i is a term of some sort s_i and $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$. We also allow subsorts: if s_2 is a subsort of s_1 we allow a terms of sort s_2 whenever a term of sort s_1 is expected. We write $sort(t)$ for the sort of term t. We define $root(t)$ to be f if $t = f(t_1, \ldots, t_n)$ and t otherwise, *i.e.* if t is either a name or a variable. We write $\mathrm{var}(t)$ and $\mathrm{names}(t)$ for the set of variables and names occurring in t, respectively. A term t is *ground* or *closed* iff $\mathrm{var}(t) = \emptyset$.

Substitutions are written $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ with domain $\mathrm{dom}(\sigma) = \{x_1, \ldots, x_n\}$. We only consider *well-sorted*, *cycle-free* substitutions. Such a σ is *closed* iff all of the t_i are closed. We let $\mathrm{var}(\sigma) = \bigcup_i \mathrm{var}(t_i)$, $\mathrm{names}(\sigma) = \bigcup_i \mathrm{names}(t_i)$, and extend the notations $\mathrm{var}(.)$ and $\mathrm{names}(.)$ to tuples and sets of terms and substitutions in the obvious way. The application of a substitution σ to a term t is written $\sigma(t) = t\sigma$ and is defined in the usual way. We also define the set of positions $pos(t)$ of a term t inductively as $pos(c) = pos(a) = pos(x) = \{\epsilon\}$ where $\mathrm{ar}(c) = s$ and $pos(f(t_1, \ldots, t_n)) = \{\epsilon\} \cup \bigcup_{1 \leq i \leq n} i \cdot pos(t_i)$. If p is a position of t then expression $t|_p$ denotes the subterm of t at the position p, *i.e.*, $t|_\epsilon = t$ and $f(t_1, \ldots, t_n)|_{i \cdot p} = t_i|_p$.

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

2.2 Deducibility and static equivalence

We use frames [3, 2] to represent sequences of messages observed by an attacker, for instance during the execution of a protocol. Formally, a *frame* is an expression $\varphi = \nu\tilde{a}.\{x_1 = t_1, \ldots, x_n = t_n\}$ where $\tilde{a}$ is a set of *bound (or restricted) names*, and for each i, t_i is a closed term of the same sort as x_i.

For simplicity, we only consider frames $\varphi = \nu\tilde{a}.\{x_1 = t_1, \ldots, x_n = t_n\}$ which restrict every name in use, that is $\tilde{a} = \mathrm{names}(t_1, \ldots, t_n)$. A name a may still be disclosed explicitly by adding a mapping $x_a = a$ to the frame. Thus we tend to assimilate such frames φ to their *underlying substitutions* $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. This substitution can also be noted $\{x_i \mapsto t_i\}_{1 \leq i \leq n}$.

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 $\varphi_1 = \{x_1 \mapsto \mathrm{enc}(k_1, k_2), x_2 \mapsto \mathrm{enc}(k_4, k_3), x_3 \mapsto k_3\}$: the name k_4 is deducible from φ_1 since $\mathrm{dec}(x_2, x_3)\varphi_1 =_{E_{\mathsf{enc}}} k_4$ but neither are k_1 nor k_2. Deducibility is not always sufficient to account for the knowledge of an attacker. For instance, it lacks partial information on secrets. We refer the reader to [2] for additional details and examples. That is why another classical notion in formal methods is *static equivalence*.

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

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

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 $s \in \mathcal{S}$; moreover, if s_2 is a subsort of s_1 we require that $[\![s_2]\!]_A \subseteq [\![s_1]\!]_A$;
- a computable function $[\![f]\!]_A : [\![s_1]\!]_A \times \ldots \times [\![s_k]\!]_A \to [\![s]\!]_A$ for each $f \in \mathcal{F}$ with $\mathrm{ar}(f) = s_1 \times \ldots \times s_k \to s$;
- an effective procedure to draw random elements from $[\![s]\!]_A$; we denote such a drawing by $x \xleftarrow{R} [\![s]\!]_A$;

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

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

Such values $\phi = \{x_1 = e_1, \ldots, x_n = e_n\}$ with $e_i \in [\![s_i]\!]_A$ are called *concrete frames*. We extend the notation $[\![.]\!]_A$ to (tuples of) closed terms in the obvious way. We also generalize the notation to terms with variables, by specifying the concrete values for all of them: $[\![.]\!]_{A, \{x_1 = e_1, \ldots, x_n = e_n\}}$.

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 $\eta \ge 0$. (This parameter η might be thought of 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 each required operation on algebras is feasible by a (uniform, probabilistic) polynomial-time algorithm in the complexity parameter η. This ensures that the concrete semantics of terms and frames is efficiently computable (in the same sense).

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 $\mathcal{A}$,

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

to be a *negligible* function of η, that is, to remain eventually smaller than any η^{-n} ($n > 0$) for sufficiently large η.

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

3 Adaptive soundness

In this section, we first recall the original notion of soundness for static equivalence which considers the presence of a passive adversary and then extend it to an adaptive adversary. We show some relations between the classical soundness and our new adaptive soundness and also provide a composition result which allows us, under some hypotheses, to prove adaptive soundness of computational algebras (A_η) from adaptive soundness of parts of (A_η).

3.1 Soundness definitions

We recall the definition of soundness of static equivalence for computational algebras introduced in [10].

Definition 3 ($\approx_E$-soundness) *Let E be an equational theory. A family of computational algebras (A_η) is* $\approx_E$-sound *if and only if for every frames φ_1, φ_2 with the same domain, $\varphi_1 \approx_E \varphi_2$ implies that $(\llbracket\varphi_1\rrbracket_{A_\eta}) \approx (\llbracket\varphi_2\rrbracket_{A_\eta})$,*

In [10], Baudet *et al.* define similar soundness notions for $=_E$ and $\vdash_E$. They also introduce the notion of *faithfulness* which intuitively corresponds to the following fact: if there exists a symbolic attack, then there also exists an efficient computational attack. In this paper we concentrate on soundness of static equivalence. As shown in [10], for many theories soundness of static equivalence implies all of the other notions. This motivates our choice, although we believe that our framework extends to these other notions without difficulties.

Baudet *et al.* also introduce a strong notion of soundness that holds without restriction on the computational power of adversaries.

Definition 4 (Unconditional $\approx_E$-soundness) *Let E be an equational theory. A family of computational algebras (A_η) is* unconditionally $\approx_E$-sound *if and only if for every frames φ_1, φ_2 with the same domain, $\varphi_1 \approx_E \varphi_2$ implies $(\llbracket\varphi_1\rrbracket_{A_\eta}) = (\llbracket\varphi_2\rrbracket_{A_\eta})$.*

Unconditional soundness stipulates that for any pairs of equivalent frames, the related distributions are equal. Hence even an adversary which is not polynomially bounded cannot distinguish these two distributions.

3.2 Adaptive security

We extend soundness of static equivalence to the adaptive setting from [28]. In $\approx_E$-soundness the adversary observes the computational value of a fixed frame whereas in this setting the adversary sees the computational value of a sequence of adaptively chosen frames.

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

$$\mathcal{O}^b_{LR,A_\eta}(t_0, t_1) = \llbracket t_b\rrbracket_{A_\eta}$$

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

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

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

Definition 5 (Adaptive soundness) *An adversary $\mathcal{A}$ is* legal *if for any sequence of queries $(t^i_0, t^i_1)_{1\le i\le n}$ made by $\mathcal{A}$ to its left-right oracle, queries are statically equivalent:*

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

A family of computational algebras (A_η) is

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

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

Proposition 1 *Let (A_η) be a family of computational algebras. If A_η is $\approx_E$-ad-sound then A_η is also $\approx_E$-sound but the converse is false in general.*

The first part of the proof is easy. Intuitively, the second part of the proof works as follows. We consider a theory with nonces and lists of bits. The theory is designed such that the adversary obtains the challenge bit if he can provide a list of bits which corresponds to the bitstring implementing a given nonce. A passive adversary has only negligible probability of success as he needs to guess the bitstring implementing the nonce. In the adaptive setting, this is however easily achieved: the adversary learns the value of a nonce through a first call to the oracle, then he submits the corresponding list of bits in a second call. The detailed proof is given in Appendix A.1. Interestingly, in the case of unconditional soundness, adaptive and non-adaptive soundness coincide.

Proposition 2 *Let (A_η) be a family of computational algebras. A_η is unconditionally $\approx_E$-ad-sound iff A_η is unconditionally $\approx_E$-sound.*

The idea is that if a family of computational algebras is unconditionally sound, then statically equivalent frames

yield the same distribution. Hence whatever the queries of the adaptive adversaries are, the distributions related to challenge bit $b = 0$ and challenge bit $b = 1$ are equal. Even an adversary whose execution time is not polynomially bounded has a zero probability to distinguish these distributions. The proof is given in Appendix A.2.

3.3 Composition result

Our objective here is to provide a composition result of the form: let Σ_1 and Σ_2 be two signatures that do not share any symbol. If A^1_η is $\approx_{E_1}$ -ad-sound and A^2_η is $\approx_{E_2}$ -ad-sound, then the composition of A^1_η and A^2_η denoted $A^1_\eta \times A^2_\eta$ is $\approx_{E_1 \cup E_2}$ -ad-sound. However this is false in general. Therefore, we provide restrictions under which composition is possible: we consider disjoint signatures as well as layered signatures.

Definition 6 (Disjoint signatures) *Let $\Sigma_1 = (\mathcal{S}_1, \mathcal{F}_1)$ and $\Sigma_2 = (\mathcal{S}_2, \mathcal{F}_2)$ be two signatures. We say that Σ_1 and Σ_2 are disjoint iff $\mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset$ and $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$.*

We denote by $\Sigma_1 \cup \Sigma_2 = (\mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{F}_1 \cup \mathcal{F}_2)$ the union of the signature Σ_1 with Σ_2.

Definition 7 (Signature combination, layered signatures) *Let $\Sigma_1 = (\mathcal{S}_1, \mathcal{F}_1)$ and $\Sigma_2 = (\mathcal{S}_2, \mathcal{F}_2)$ be two disjoint signatures. We say that a subsort relation S is a* signature combination *for Σ_1 and Σ_2 if $\mathsf{S} \subseteq \mathcal{S}_2 \times \mathcal{S}_1$.*

If S is a signature combination for Σ_1 and Σ_2, we also say that $\Sigma = \Sigma_1 \cup \Sigma_2$ is a $(\Sigma_1, \Sigma_2)_\mathsf{S}$-layered signature.

Intuitively, if a signature is layered then a constructor of $\mathcal{F}_1$ never occurs under a constructor of $\mathcal{F}_2$ and S defines which sorts of Σ_2 can be used as subsort of Σ_1.

Given such a $(\Sigma_1, \Sigma_2)_\mathsf{S}$-layered signature Σ and a term t over Σ we define the set of Σ_1 positions of t

$$pos_{\Sigma_1}(t) = \{p \mid p \in pos(t), sort(t|_p) \in \mathcal{S}_1\}$$

and the set of Σ_2 minimal positions of t

$$pos^\star_{\Sigma_2}(t) = \{p \mid \begin{array}{l} p \in pos(t), sort(t|_p) \in \mathcal{S}_2, \\ p = p' \cdot i \Rightarrow sort(t|_{p'}) \notin \mathcal{S}_2\} \end{array}$$

As an example let us consider a theory with a symmetric encryption schemes and a pseudo-random generator. The signature Σ_1 is composed of a sort Data and of two symbols enc and dec, both of arity $\mathrm{Data} \times \mathrm{Data} \to \mathrm{Data}$. Signature Σ_2 contains one sort Rand and a symbol prg (for pseudo-random generator) of arity $\mathrm{Rand} \to \mathrm{Rand}$. We define the signature combination S that contains a single element $(\mathrm{Rand}, \mathrm{Data})$: this models that elements of sort Rand can be used either as keys or as plaintext. We have that Σ_1 and Σ_2 are disjoint, and $\Sigma = \Sigma_1 \cup \Sigma_2$ is $(\Sigma_1, \Sigma_2)_\mathsf{S}$-layered. Given the term $t = \mathrm{enc}(\mathrm{enc}(\mathrm{prg}(r), k), \mathrm{prg}(\mathrm{prg}(r')))$ where $k \in \mathrm{Data}$ and $r, r' \in \mathrm{Rand}$, t is indeed a valid term of Σ. However, the term $t' = \mathrm{prg}(\mathrm{enc}(r, k))$ is not a term of Σ as it is not well sorted. We have that $pos_{\Sigma_1}(t) = \{\epsilon, 1, 12\}$ and $pos^\star_{\Sigma_2} = \{11, 2\}$.

Definition 8 (Hybrid functions) *Let Σ_1 and Σ_2 be two disjoint signatures and S a signature combination such that $\Sigma = \Sigma_1 \cup \Sigma_2$ is $(\Sigma_1, \Sigma_2)_\mathsf{S}$-layered. Let E_1 and E_2 be equational theories over Σ_1 and Σ_2 respectively. A (E_1, E_2)-*hybrid function *for a set F of pairs of frames is a function σ from lists of terms over Σ to terms over Σ such that:*

- *for any frame φ occurring in F, $\varphi \approx_{E_1} \sigma(\varphi)$ where we naturally extended σ over frames by $\sigma(\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}) = \{x_1 \mapsto \sigma([t_1]), \ldots, x_n \mapsto \sigma([t_1 \ldots t_n])\}$;*

- *for any $(\varphi, \varphi') \in F$, if $\varphi \approx_{E_1 \cup E_2} \varphi'$ then let $\varphi = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ and $\varphi' = \{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$. We have that for all i in $[1, n]$,*

  - *$pos_{\Sigma_1}(\sigma([t_1 \ldots t_i])) = pos_{\Sigma_1}(\sigma([u_1 \ldots u_i])) = P$ and for any $p \in P$*

    $$root(\sigma([t_1 \ldots t_i])|_p) = root(\sigma([u_1 \ldots u_i])|_p)$$

  - *$pos^\star_{\Sigma_2}(\sigma([t_1 \ldots t_i])) = pos^\star_{\Sigma_2}(\sigma([u_1 \ldots u_i])) = Q$ and we have that*

    $$\begin{array}{c} \{x_q \mapsto \sigma([t_1 \ldots t_i])|_q\}_{q \in Q} \\ \approx_{E_2} \\ \{x_q \mapsto \sigma([u_1 \ldots u_i])|_q\}_{q \in Q} \end{array}$$

Moreover σ has to be computable in polynomial time (in its input).

Adaptive soundness may not hold on all frames, although it holds on a subset of well-formed frames, *e.g.*, when considering encryption one typically discards all frames that contain key cycles. Therefore we say that an abstract algebra A_η is $\approx_E$*-ad-sound for a set F of pair of frames* if the advantage $\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A}, A_\eta}(\eta)$ of any polynomial-time legal adversary $\mathcal{A}$, whose sequence of queries $(t^i_0, t^i_1)_i$ verifies that the pair $(\{x_i \mapsto t^i_0\}_i, \{x_i \mapsto t^i_1\}_i)$ is in F, is negligible. We will typically show soundness for the set of all pairs of "well-formed" frames (the notion of well-formed frames depends on the particular equational theory).

Proposition 3 (Composition) *Let Σ_1 and Σ_2 be two disjoint signatures and S be a signature combination for Σ_1 and Σ_2. Let E_1 and E_2 be equational theories over Σ_1 and Σ_2 respectively. We consider a family of computational algebras (A^1_η) for Σ_1 and another family (A^2_η) for Σ_2 respecting S, i.e. $(s_2, s_1) \in \mathsf{S}$ implies that $[\![s_2]\!]_{A^2_\eta} \subseteq [\![s_1]\!]_{A^1_\eta}$.*

Let F be a set of pair of frames over $\Sigma_1 \cup \Sigma_2$ and σ be a (E_1, E_2)-hybrid function for F. If $A^1_\eta \times A^2_\eta$ is $\approx_{E_1}$-ad-sound for $G = \{(\varphi, \sigma(\varphi)) \mid \varphi \text{ occurs in } F\}$ and A^2_η is $\approx_{E_2}$-ad-sound for frames on Σ_2, then $A^1_\eta \times A^2_\eta$ is $\approx_{E_1 \cup E_2}$-ad-sound for F.

The detailed proof of this proposition is given in Appendix A.3. The idea is that if an adversary $\mathcal{A}$ against $E_1 \cup E_2$-ad-soundness queries his oracle with a pair of frames (φ, φ') in F then it is possible to build an adversary $\mathcal{B}_1$ against E_1-ad-soundness who submits $(\varphi, \sigma(\varphi))$ to his oracle, an adversary $\mathcal{B}_2$ against E_2-ad-soundness who submits $(\sigma(\varphi), \sigma(\varphi'))$ and an adversary $\mathcal{B}_3$ against E_1-ad-soundness who submits $(\sigma(\varphi'), \varphi')$ such that the advantages of $\mathcal{A}$, $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ are related. This composition result will be useful in Section 4 when composing encryption with modular exponentiation and XOR. It allows us to show soundness for these joint theories avoiding nearly any additional proofs at the computational level.

4 Adaptively sound theories

We now present adaptive soundness results for several equational theories: symmetric encryption (which is adaptively sound under IND-CPA), exclusive or (unconditionally adaptively sound) and modular exponentiation (adaptively sound under DDH). We also consider composed theories: symmetric encryption and modular exponentiation as well as symmetric encryption and exclusive or.

4.1 Symmetric encryption

We consider the case of probabilistic symmetric encryption and try to be as close as possible to the models from [4] and from [28]. Hence we assume that the implementation of the symmetric encryption scheme is semantically secure [24] and use a relevant formal theory.

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

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

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

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

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

Well-formed frames and adversaries. The importance of key cycles was already described in [4]. In general IND-CPA is not sufficient to prove any soundness result in the presence of key cycles. Thus, as in numerous previous work, we forbid the formal terms to contain such cycles. Let $\prec$ be a total order among keys. A *frame φ is acyclic for $\prec$* if for any subterm $\{t\}_k$ of φ, if k' occurs in t then $k' \prec k$. (Another possibility to handle key cycles is to consider stronger computational requirements like Key Dependent Message – KDM – security as done in [5].) Moreover as noted in [28], selective decommitment [23] can be a problem. The classical solution to avoid this problem is to require keys to be sent *before* being used to encrypt a message or they must never appear as a plaintext. We say that a *frame $\varphi = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is well-formed for $\prec$* if

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

We also say that an *adversary is well-formed for $\prec$* if the sequence of queries $(t^i_0, t^i_1)_{1 \le i \le n}$ that he makes to his oracle yields two well-formed frames $\{x_1 \mapsto t^1_0, \ldots, x_n \mapsto t^n_0\}$ and $\{x_1 \mapsto t^1_1, \ldots, x_n \mapsto t^n_1\}$ for $\prec$.

Concrete model. 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 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 $c \leftarrow \mathcal{E}(k, s)$ then it is required that $\mathcal{D}(c) = s$.

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

Semantic security. In this paper we use schemes that satisfy length-concealing semantic security. 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. The oracle returns ciphertext $\mathcal{E}(bs_b, k)$. The goal of the adversary $\mathcal{A}$ is to guess the value of bit b. His advantage is defined as:

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

Encryption scheme $\mathcal{SE}$ is IND-CPA secure if the advantage of any adversary $\mathcal{A}$ is negligible in η. The difference with the standard notion of semantic security is that we require that the scheme hides the length of the plaintext (and therefore we do not restrict bs_0 and bs_1 to have equal length). By abuse of notation we call the resulting scheme also IND-CPA secure.

We also describe a variant of IND-CPA security, IND-CPA′, which models non-adaptive adversaries. The left-right encryption oracle $LR'^b_{\mathcal{SE}}$ takes as input a list of pairs of bit-strings (bs^i_0, bs^i_1) for i in $[1, n]$ and returns the list of ciphertexts $\mathcal{E}(bs^i_b, k)$ for i in $[1, n]$. This oracle can only be queried once. The adversary can observe multiple encryptions but he is not allowed to chose them adaptively. The advantage of an adversary is defined in a similar way as above, replacing $LR^b_{\mathcal{SE}}$ by $LR'^b_{\mathcal{SE}}$. A symmetric encryption scheme is said to be IND-CPA′ if the advantage of any polynomial time adversary $\mathcal{A}$ is negligible in η.

These two notions of semantic security can be related through the following proposition whose proof is delayed to Appendix B.1.

Proposition 4 *Let $\mathcal{SE}$ be a symmetric encryption scheme. If $\mathcal{SE}$ is* IND-CPA*, then $\mathcal{SE}$ is* IND-CPA′*. However $\mathcal{SE}$ can be* IND-CPA′ *without being* IND-CPA*.*

The first implication is trivial. In order to obtain the converse, we consider an encryption scheme $\mathcal{SE}'$ with a "point of weakness" m (for example the encryption of m is 0 and each ciphertext divulges the point of weakness m). It is easy for an adversary against IND-CPA to exploit this weakness whereas an adversary against IND-CPA′ has only negligible probability to find the weakness as he is not adaptive.

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

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

The proof of this proposition (given in Appendix B.2) uses a similar hybrid argument as the one used by Micciancio and Panjwani in [28]. Results of this section are summed up in the following table.

Note that the relations between adaptive and non-adaptive soundness have not been detailed formally. They can be proven using the same counter-example as in the proof of Proposition 4.

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

4.2 Exclusive OR

We study the adaptive soundness problem for the usual theory and implementation of the Exclusive Or (XOR) in the same model as given in [10]. The symbolic model $\Sigma_\oplus$ consists of a single sort $\mathrm{Data}_\oplus$, an infinite number of names, the infix symbol $\oplus : \mathrm{Data}_\oplus \times \mathrm{Data}_\oplus \to \mathrm{Data}_\oplus$ and two constants $0_\oplus, 1_\oplus : \mathrm{Data}_\oplus$. Terms are equipped with the equational theory $E_\oplus$ generated by:

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

As an implementation, we define the computational algebras A_η: the concrete domain $[\![\mathrm{Data}_\oplus]\!]_{A_\eta}$ is $\{0,1\}^\eta$

equipped with the uniform distribution; $\oplus$ is interpreted by the usual XOR function over $\{0,1\}^\eta$, $[\![0_\oplus]\!]_{A_\eta} = 0^\eta$, $[\![1_\oplus]\!]_{A_\eta} = 1^\eta$. This implementation of XOR enjoys unconditional adaptive soundness with respect to $\approx_{E_\oplus}$.

Proposition 6 *The usual implementation for the XOR theory is unconditionally $\approx_{E_\oplus}$-ad-sound.*

Proof: In [10], the usual implementation for the XOR theory is proven to be unconditionally $\approx_{E_\oplus}$-sound. Proposition 2 allows us to conclude. ∎

4.3 Modular exponentiation

As a third application, we study soundness of modular exponentiation. The cryptographic assumption we make is that the *Decisional Diffie-Hellman* (DDH) problem is difficult: even when given g^x and g^y, it is difficult for any feasible computation to distinguish between g^{xy} and g^r, when x, y and r are selected at random. The original Diffie-Hellman protocol [22] has been used as a building block for several key agreement protocols that are widely used in practice (e.g. SSL/TLS and Kerberos V5) as well as for group key exchange protocols such as AKE1 [14] or the Burmester-Desmedt protocol [16]. Our objective is to be able to study such protocols.

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

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

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

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

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

We put two restrictions on formal terms: products have to be *power-free*, *i.e.*, x^n is forbidden for $n > 1$, and products must not contain more than l elements for some fixed bound l, i.e. $x_1 \cdot \ldots \cdot x_n$ is forbidden for $n > l$. Both restrictions come from the DDH assumption and seem difficult to avoid [15]. Furthermore we are only interested in frames using terms of sort G. Any frame containing only terms of sort G can be rewritten as $\{x_1 \mapsto \exp(p_1), ..., x_n \mapsto \exp(p_n)\}$ by orienting the last equation form left to right. For such frames there is an immediate characterization of static equivalence. Two frames are statically equivalent if they satisfy the same linear equations.

Proposition 7 *We have that*

$$\{x_1 \mapsto \exp(p_1), ..., x_n \mapsto \exp(p_n)\} \approx_{E_{\mathsf{DH}}} \{x_1 \mapsto \exp(q_1), ..., x_n \mapsto \exp(q_n)\}$$

iff for any sequence of integer $a_0, a_1, \ldots a_n$

$$a_0 + \sum_{i=1}^{n} a_i p_i = 0 \Leftrightarrow a_0 + \sum_{i=1}^{n} a_i q_i = 0$$

The proof is given in Appendix B.3. This characterization can be used to decide static equivalence efficiently.

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

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

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

is negligible in η. In the remainder, we generally suppose that for any η there is a unique group given by IG. We show that the DDH assumption is necessary and sufficient to prove adaptive soundness.

Proposition 8 *Let (A_η) be a family of computational algebras. (A_η) is $\approx_{E_{\mathsf{DH}}}$-sound iff (A_η) satisfies the* DDH *assumption. (A_η) is $\approx_{E_{\mathsf{DH}}}$-ad-sound iff (A_η) satisfies the* DDH *assumption.*

The proof of this result (given in Appendix B.4) uses an adaptive variant of DDH called 3DH: it generalizes several previously used variants of DDH. The main difficulty in this proof consists in relating DDH and 3DH.

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

$$\begin{array}{rcl} \approx_{E_{\mathsf{DH}}}\text{-ad-sound} & \iff & \mathsf{DDH} \\ & & \Updownarrow \\ \approx_{E_{\mathsf{DH}}}\text{-sound} & \iff & \mathsf{DDH} \end{array}$$

4.4 Composing encryption with exponentiation

We illustrate our composition result (Proposition 3) by establishing a joint soundness result for symmetric encryption and modular exponentiation.

Symbolic model. We consider an equational theory E containing both E_{DH} and E_{sym}. Let Σ_1 be the signature for symmetric encryption and Σ_2 be the signature for modular exponentiation, then signature $\Sigma = \Sigma_1 \cup \Sigma_2$ is $(\Sigma_1, \Sigma_2)_{\mathsf{S}}$-layered where S contains only one element (G, Data).

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

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

Concrete model. The concrete model is given by the models for symmetric encryption and modular exponentiation. However, exponentiations can be used as symmetric keys in our symbolic model which needs to be reflected in the concrete model. The family of computational algebras (A_η) giving the concrete semantics is parameterized by a symmetric encryption scheme $\mathcal{SE}$ and an instance generator IG. We require that the key generation algorithm of $\mathcal{SE}$ randomly samples an element of $IG(\eta)$. Giving an IND-CPA encryption scheme $\mathcal{SE}'$, it is possible to build another IND-CPA encryption scheme $\mathcal{SE}$ which indeed uses such a key generation algorithm. This is achieved by using a *key extractor* algorithm Kex [18]. This algorithm (usually a universal hash function used with the entropy smoothing theorem) is used to transform group elements into valid keys for $\mathcal{SE}'$. Its main characteristic is that applying Kex to a randomly sampled element of a group created by IG produces the same distribution as the one given by the key generation algorithm of $\mathcal{SE}'$. Then the new encryption and decryption algorithms of $\mathcal{SE}$ apply the Kex algorithm to the group element which is used as key. This produces a symmetric key which can be used with the encryption and decryption algorithms of $\mathcal{SE}'$.

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

Soundness is proven by applying Proposition 3. The proof is detailed in Appendix B.5.

Proposition 9 *Let $\prec$ be a total order between keys and exponentiations. Let (A_η) be an EE-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

4.5 Composing encryption with XOR

As another illustration of our composition result, we prove a joint soundness result for symmetric encryption and XOR.

Symbolic model. We consider an equational theory E containing both $E_\oplus$ and E_{sym}. Let $\Sigma_1 = \Sigma_{\mathrm{sym}}$ be the signature for symmetric encryption and $\Sigma_2 = \Sigma_\oplus$ be the signature for exclusive OR. Then signature $\Sigma = \Sigma_1 \cup \Sigma_2$ is $(\Sigma_1, \Sigma_2)_{\mathsf{S}}$-layered where S contains only one element $(\mathrm{Data}_\oplus, \mathrm{Data})$.

Well-formed frames. Let $\prec$ be a total order between keys and terms of sort $\mathrm{Data}_\oplus$. A frame $\varphi = \{x_i \mapsto t_i\}_{1 \le i \le n}$ (on Σ) is well-formed for $\prec$ if the following conditions are verified. Let $X = \bigcup_{1 \le i \le n} \left\{ t_i|_p \mid p \in pos^\star_{\Sigma_2}(t_i) \right\}$.

- φ does not contain function symbols dec, tenc, tpair, π_l or π_r and only terms of sort $\mathrm{Data}_\oplus$ and names are used at key positions.
- For any $x \in X$ used at a key position, there does not exist a set $\{x_1, \ldots, x_i\} \subseteq X \cup \{1\}$, disjoint from $\{x\}$, such that $x =_{E_\oplus} x_1 \oplus \ldots \oplus x_i$.

- For any subterm $\{t\}_{t'}$ of φ, if t'' is a subterm of t which is a name of sort Data or an element of X then $t'' \prec t'$.

Concrete model. The concrete model is given by the models for symmetric encryption and exclusive OR. However, as in the combination of encryption with exponentiation, we need to reflect that nonces can be used as keys. The family of computational algebras (A_η) giving the concrete semantics is parameterized by a symmetric encryption scheme $\mathcal{SE}$. The XOR part uses the same implementation as in Section 4.2. We require that the key generation algorithm of $\mathcal{SE}$ consists in randomly sampling an element of $[0,1]^\eta$. The family of computational algebras (A_η) is said *EX-secure* if the encryption scheme $\mathcal{SE}$ is secure against IND-CPA and uses a key generation algorithm as described above.

Soundness can be proven by applying Proposition 3. The proof is detailed in Appendix B.6.

Proposition 10 *Let $\prec$ be a total order between keys and terms of sort $\mathrm{Data}_\oplus$. Let (A_η) be an EX-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

5 Analysis of dynamic group key exchange

Micciancio and Panjwani exemplified their adaptive soundness result from [28] on multicast protocols. We propose another application: *dynamic group key exchange protocols* (DKE). Our objective is to provide a framework for classical DKE protocols such as the AKE1 protocol from [14]. Therefore we only define security for protocols using only modular exponentiation: we consider a subtheory E of E_{DH} (Section 4.3) without the $+$, $-$, 1_R and 0_R symbols and their related equations. We remove these symbols in order to keep the symbolic security notion as simple as possible. However our definitions and soundness results can be adapted to other equational theories (e.g. symmetric encryption joint with modular exponentiation).

5.1 Dynamic group protocols

We take a simple model for DKE in the adaptive setting. A DKE protocol is described by four operations which specify the protocol. We suppose that this specification is given by four polynomial-time algorithms $(\mathcal{S}, \mathcal{J}, \mathcal{L}, \mathcal{K})$:

- $\mathcal{S}$ initializes a new group. The algorithm takes as an input a list of users and outputs the internal state s_0 of the protocol as well as a list of formal terms which model the messages that have been exchanged during the setup phase.
- $\mathcal{J}$ and $\mathcal{L}$ take as input the state of the protocol s and a list of users U_1 to U_n (to be respectively added to or suppressed from the group) and output the updated state of the protocol s' as well as a list of formal terms representing message exchanges.
- $\mathcal{K}$ takes as input the state of the group s and outputs a formal term representing the shared key of the group.

The internal state of the protocol can be thought of as the internal state of the four algorithms that describe the protocol.

We partition the set of names of sort R according to the users: n_i^j, $j \in \mathbb{N}$, are the nonces generated by user U_i. We require that the formal term output by $\mathcal{K}$ only uses nonces for users that are currently in the group.

5.2 Security in the symbolic model

In our symbolic setting, the security property is expressed through reachability in a transition system. We represent the states of this transition system as a triple $\langle L, C, T\rangle$ where

- L is the list of users that are currently in the group;
- C is the set of corrupted users;
- T is the list of formal terms sent during the protocol execution.

We suppose that the internal state of the protocol can be recovered from the state $\langle L, C, T\rangle$ and tend to assimilate these two notions of state. We now describe the possible transitions. For convenience, we use set notations for manipulating lists.

1. $\langle \emptyset, C, \emptyset\rangle \xrightarrow{\mathsf{c}(U)} \langle \emptyset, C \cup \{U\}, \emptyset\rangle$: corruption of user U.
2. $\langle \emptyset, C, \emptyset\rangle \xrightarrow{\mathsf{s}(\mathcal{U})} \langle \mathcal{U}, C, T\rangle$: setup of the group, *i.e.*, $\langle \mathcal{U}, C, T\rangle$ is computed by $\mathcal{S}(\langle \emptyset, C, \emptyset\rangle, \mathcal{U})$, where $\mathcal{U}$ is a list of users.
3. $\langle L, C, T\rangle \xrightarrow{\mathsf{j}(\mathcal{U})} \langle L \cup \mathcal{U}, C, T\rangle \cup T'$: join of users in the list $\mathcal{U}$, *i.e.*, $\langle L \cup \mathcal{U}, C, T \cup T'\rangle$ is computed by $\mathcal{J}(\langle L, C, T\rangle, \mathcal{U})$.
4. $\langle L, C, T\rangle \xrightarrow{\mathsf{l}(\mathcal{U})} \langle L \setminus \mathcal{U}, C, T \cup T'\rangle$: exclusion of the users in the list $\mathcal{U}$, *i.e.*, $\langle L \setminus \mathcal{U}, C, T \cup T'\rangle$ is computed by $\mathcal{L}(\langle L, C, T\rangle, \mathcal{U})$.

To simplify things up, we consider a static corruption model, *i.e.*, corruption transitions only occur at the beginning of the protocol. Then the setup transition has to occur and after only leave and join transitions are allowed. A DKE protocol is secure if it is impossible for an adversary to get any bit of information on the group key when no corrupted users are in the group.

Definition 9 *A DKE protocol is* symbolically secure *if for any state* $\langle L, C, T = \{t_1, ..., t_n\}\rangle$ *such that* $C \cap L = \emptyset$ *which is reachable from* $\langle \emptyset, \emptyset, \emptyset\rangle$ *we have that*

$$\{x_1 \mapsto t'_1, ..., x_n \mapsto t'_n, y \mapsto \mathcal{K}(\langle L, C, T\rangle)\} \approx_E \{x_1 \mapsto t'_1, ..., x_n \mapsto t'_n, y \mapsto \exp(r)\}$$

where r *is a fresh nonce,* $N = \{n_i^j \mid U_i \in C\}$ *and* t'_i *is as* t_i *but nonces from* N *have been removed, i.e. if* $t = \exp(m_1 \cdot \ldots \cdot m_\ell)$ *then* $t' = \exp(m'_1 \cdot \ldots \cdot m'_{\ell'})$ *where* $\{m'_1, \ldots, m'_{\ell'}\} = \{m_1, \ldots, m_\ell\} \setminus N$.

5.3 Security in the concrete model

We use a simplified version of the security model from [14]: some oracles in [14] are not useful anymore in the adaptive setting. Let $(\mathcal{S}, \mathcal{J}, \mathcal{L}, \mathcal{K})$ be a DKE and (A_η) a family of computational algebras. Adversary $\mathcal{A}$ interacts with the group via the following five oracles which store the current state s of the group and use a challenge bit b.

- $\mathsf{Setup}(U_1, ..., U_n)$: this query initializes the group using $\mathcal{S}(U_1, ..., U_n)$ which produces the new state s and a list of formal terms t_1 to t_m. $\mathcal{A}$ is given $[\![t_i]\!]_{A_\eta}$ for any i in $[1, m]$.
- $\mathsf{Join}(U_1, ..., U_n)$: $\mathcal{A}$ asks users U_1 to U_n to join the group. Algorithm $\mathcal{J}(s, U_1, ..., U_n)$ is executed and outputs the new state s and a list of formal terms t_1 to t_m. $\mathcal{A}$ is given $[\![t_i]\!]_{A_\eta}$ for any i in $[1, m]$.
- $\mathsf{Leave}(U_1, ..., U_n)$: $\mathcal{A}$ asks users U_1 to U_n to leave the group. Algorithm $\mathcal{L}(s, U_1, ..., U_n)$ is executed and outputs the new state s and a list of formal terms t_1 to t_m. $\mathcal{A}$ is given $[\![t_i]\!]_{A_\eta}$ for any i in $[1, m]$.
- $\mathsf{Corrupt}(U)$: $\mathcal{A}$ corrupts user U, all the nonces generated by U are given to $\mathcal{A}$. As $\mathcal{A}$ works in polynomial time, it is only necessary to give him a polynomial number of values.
- Test: $\mathcal{A}$ either receives the key of the group (output by $\mathcal{K}(s)$) if $b = 1$ or a random key if $b = 0$ and has to decide which is the case. This oracle can only be queried once.

As we consider a static corruption model, queries to the Corrupt oracle have to be done before all further queries. Then the Setup oracle is called and after that the adversary interleaves queries to the Join and Leave oracles. The adversary makes a final call to the Test oracle. Let $\mathcal{O}_b$ denote the oracles with challenge bit b. The advantage of an adversary $\mathcal{A}$ is given by:

$$\mathrm{Adv}^{(\mathcal{S},\mathcal{J},\mathcal{L},\mathcal{K})}_{\mathcal{A},A_\eta}(\eta) = \mathbb{P}\left[\mathcal{A}^{\mathcal{O}_1} = 1\right] - \mathbb{P}\left[\mathcal{A}^{\mathcal{O}_0} = 1\right]$$

The DKE is said to be *secure in the concrete model* if the advantage of any adversary is negligible in η.

5.4 Soundness result

Our symbolic model for DKE is computationally sound: if a DKE algorithm is secure in the symbolic model, then it is secure in the computational model, provided that static equivalence is adaptively sound (remember that we consider only modular exponentiation hence static equivalence is adaptively sound under DDH).

Proposition 11 *Let* (A_η) *be a family of computational algebras and* $\Pi = (\mathcal{S}, \mathcal{J}, \mathcal{L}, \mathcal{K})$ *be a DKE. If* (A_η) *is* $\approx_E$*-ad-sound and* Π *is secure in the symbolic model, then* Π *is secure in the concrete model.*

The proof of this result is given in Appendix C: access to the oracles are mapped to transitions in the symbolic setting. As the protocol is symbolically secure, the sequence of exchanged terms are statically equivalent when considering the real key and when considering a randomly sampled key. Adaptive soundness is then used to obtain concrete security.

6 Conclusion

In this paper we defined a framework for reasoning about the soundness of equational theories in the presence of adaptive adversaries. While an adaptive adversary cannot intercept and inject messages it is allowed to alter the execution flow of a protocol. We give definitions of what it means for a computational algebra to be adaptively sound with respect to static equivalence and study several equational theories: symmetric encryption, XOR and modular exponentiation. We also provide a combination technique to show soundness for the joint theories of encryption and XOR as well as encryption and modular exponentiation. These results are significant as they enable cryptographically sound symbolic proofs for practical protocols based on Diffie-Hellman key exchange. Finally we demonstrate the usefulness of adaptive soundness by giving a sound symbolic model for the analysis of DKE protocols.

This work opens the possibility for studying other interesting equational theories in an adaptive setting, such as the theory used in [1] in the context of offline guessing attacks. A natural future work is to use the symbolic model for DKE protocols defined in this paper on case studies. Finally an ambitious extension is the soundness in the presence of a both active and adaptive adversary.

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 V. Cortier. Deciding knowledge in security protocols under equational theories. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 46–58, 2004.

[3] M. Abadi and C. Fournet. Mobile values, new names, and secure communications. In *Proc. 28th Annual ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.

[4] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS'00)*, volume 1872 of *Lecture Notes in Computer Science*. Springer, 2000.

[5] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *Lecture Notes in Computer Science*, pages 374–396. Springer, 2005.

[6] M. Arnaud, V. Cortier, and S. Delaune. Combining algorithms for deciding knowledge in security protocols. Technical report, INRIA, 2007. `http://www.loria.fr/~cortier/Papiers/combination.pdf`.

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

[8] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *Proc. 10th ACM Conference on Computer and Communications Security (CCS'03)*, 2003.

[9] 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)*, Lecture Notes in Computer Science. Springer, 2006. To appear.

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

[11] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *Advances in Cryptology - EUROCRYPT'00, Proc. International Conference on the Theory and Application of Cryptographic Techniques*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274. Springer, 2000.

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

[13] B. Blanchet. A computationally sound mechanized prover for security protocols. In *Proc. 27th IEEE Symposium on Security and Privacy (SSP'06)*, pages 140–154. IEEE Computer Society Press, 2006.

[14] E. Bresson, O. Chevassut, and D. Pointcheval. Provably authenticated group Diffie-Hellman key exchange – the dynamic case. In *Advances in Cryptology - ASIACRYPT '01, Proc. 7th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2248 of *Lecture Notes in Computer Science*, pages 290–309. Springer, 2001.

[15] E. Bresson, Y. Lakhnech, L. Mazaré, and B. Warinschi. A generalization of DDH with applications to protocol analysis and computational soundness. Submitted, an online version is available at `http://www.lsv.ens-cachan.fr/~mazare/BLMW.pdf`, 2007.

[16] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system (extended abstract). In *Advances in Cryptology - EUROCRYPT'94, Proc. Workshop on the Theory and Application of Cryptographic Techniques*, volume 950 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 1994.

[17] R. Canetti and J. Herzog. Universally composable symbolic analysis of mutual authentication and key-exchange protocols (extended abstract). In *Proc. 3rd Theory of Cryptography Conference (TCC'06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 380–403. Springer, 2006.

[18] 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/`.

[19] V. Cortier, S. Delaune, and P. Lafourcade. A Survey of Algebraic Properties Used in Cryptographic Protocols. *Journal of Computer Security*, To appear, 2005.

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

[21] A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic Polynomial-time Semantics for a Protocol Security Logic. In *Proc. of 32nd International Colloquium on Automata, Languages and Programming, ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2005. Lisboa, Portugal.

[22] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[23] C. Dwork, M. Naor, O. Reingold, and L. J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

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

[25] J. Herzog. The Diffie-Hellman key-agreement scheme in the strand-space model. In *Proc. 16th IEEE Computer Science Foundations Workshop (CSFW'03)*, pages 234–247, 2003.

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

[27] J. Katz and M. Yung. Scalable protocols for authenticated group key exchange. In *Advances in Cryptology - CRYPTO 2003, Proc. 23rd Annual International Cryptology Conference*, volume 2729 of *Lecture Notes in Computer Science*, pages 110–125. Springer, 2003.

[28] D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Proc. 2nd Theory of cryptography conference (TCC'05)*, volume 3378 of *Lecture Notes in Computer Science*, pages 169–187. Springer, 2005.

[29] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proc. 1st Theory of Cryptography Conference (TCC'04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[30] C. Sprenger, M. Backes, D. Basin, B. Pfitzmann, and M. Waidner. Cryptographically sound theorem proving. In *Proc. 19th IEEE Computer Science Foundations Workshop (CSFW'06)*, pages 153–166, 2006.

A Proofs for Section 3

A.1 Proof of Proposition 1

Proposition 1 *Let (A_η) be a family of computational algebras. If A_η is $\approx_E$-ad-sound then A_η is also $\approx_E$-sound but the converse is false in general.*

The first implication is easy to prove: let $\varphi = \{x_i \mapsto t_i\}_{1\leq i\leq n}$ and $\varphi' = \{x_i \mapsto u_i\}_{1\leq i\leq n}$ be two frames such that $\varphi \approx_E \varphi'$. Let $\mathcal{A}$ be an adversary against soundness for φ and φ', (*i.e.*, $\mathcal{A}$ tries to distinguish implementations of φ from implementations of φ') we build an adversary $\mathcal{B}$ against adaptive soundness which uses his oracle to obtain an implementation of φ or φ'.

Adversary $\mathcal{B}^{\mathcal{O}_{LR}}$:
  for i **from** 1 **to** n
    $bs_i \leftarrow \mathcal{O}_{LR}(t_i, u_i)$
  return $\mathcal{A}\left(\{x_i \mapsto bs_i\}_{1\leq i\leq n}\right)$

When the challenge bit of $\mathcal{B}$ is 0, $\mathcal{A}$ is given an implementation of φ whereas when it is 1, $\mathcal{A}$ is given an implementation of φ'. The advantage of $\mathcal{B}$ is:

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{B},A_\eta}(\eta) &= \left|\mathbb{P}\left[\mathcal{B}^{\mathcal{O}^1_{LR,A_\eta}} = 1\right] - \mathbb{P}\left[\mathcal{B}^{\mathcal{O}^0_{LR,A_\eta}} = 1\right]\right| \\
&= \left|\mathbb{P}\left[\widehat{\phi} \leftarrow [\![\varphi]\!]_{A_\eta} \ : \ \mathcal{A}(\eta, \widehat{\phi})\right] - \right. \\
&\qquad \left. \mathbb{P}\left[\widehat{\phi} \leftarrow [\![\varphi']\!]_{A_\eta} \ : \ \mathcal{A}(\eta, \widehat{\phi})\right]\right| \\
&= \mathrm{Adv}^{\mathsf{IND}}_{\mathcal{A}}(\eta, [\![\varphi]\!]_{A_\eta}, [\![\varphi']\!]_{A_\eta})
\end{aligned}$$

As we assume (A_η) to be $\approx_E$-ad-sound, the advantage of $\mathcal{B}$ is negligible, so we obtain that $[\![\varphi]\!]_{A_\eta}$ and $[\![\varphi']\!]_{A_\eta}$ are indistinguishable and (A_η) is $\approx_E$-sound.

To prove that the converse statement does not hold, we give a counter example based on the following theory. We consider four sorts, Nonce, Bit, Bool and BS where Bit is a subtype of BS, and the following symbols:

$$\begin{aligned}
0, 1 &: \mathrm{Bit} \\
\mathrm{cons} &: \mathrm{Bit} \times \mathrm{BS} \to \mathrm{BS} \\
\mathrm{eq} &: \mathrm{BS} \times \mathrm{Nonce} \to \mathrm{Bool}
\end{aligned}$$

For soundness, we only consider frames which contain terms of sort Nonce or of sort Bool. In particular we assume that the frame does not contain any names that are not of sort Nonce. We do not consider any equational theory.

As an implementation, we define the computational algebras A_η: the concrete domain $[\![\mathrm{Nonce}]\!]_{A_\eta}$ is $\{0,1\}^\eta$ equipped with the uniform distribution; the constants 0 and 1 are respectively interpreted with 0 and 1. The implementation of sort Bool contains the same values 0 and 1. cons is interpreted by bit concatenation, $[\![\mathrm{eq}]\!](bs, N)$ outputs 1 if

the computational interpretations of bs and N are the same, 0 otherwise. As we do not consider any equational theory, $\{x_i \mapsto t_i\}_{1\leq i\leq n}$ and $\{x_i \mapsto u_i\}_{1\leq i\leq n}$ are statically equivalent iff there exists a renaming of nonces σ such that for any i, $t_i = N$ and $u_i = N\sigma$ or $t_i = \mathrm{eq}(t, N)$ and $u_i = \mathrm{eq}(t', N')$. We easily build an adversary $\mathcal{A}$ against adaptive soundness that has an advantage of 1: the first query of $\mathcal{A}$ is (N, N). $\mathcal{A}$ obtains the bit-string value bs of N and produces term t of type BS which has the same implementation. $\mathcal{A}$ also produces a term t' equal to t except that its first bit has been flipped. Then $\mathcal{A}$ queries his oracle with $(\mathrm{eq}(t', N), \mathrm{eq}(t, N))$. If the oracle answer is 1, $\mathcal{A}$ outputs 1, else $\mathcal{A}$ outputs 0, in both cases $\mathcal{A}$ has correctly guessed his challenge bit. Hence his advantage is 1.

However we prove that (A_η) is $\approx_E$-sound, *i.e.*, the advantage of any adversary $\mathcal{A}$ in distinguishing two statically equivalent frame is negligible. We consider two statically equivalent frames φ and φ' Without loss of generality (renaming of nonces and variables), φ has the form:

$$\begin{array}{ll}\{ & x_1 \mapsto N_1, \ldots, x_k \mapsto N_k,\\ & y_1 \mapsto \mathrm{eq}(bs_1, M_1), \ldots, y_\ell \mapsto \mathrm{eq}(bs_\ell, M_\ell)\}\end{array}$$

and φ' has the form:

$$\begin{array}{ll}\{ & x_1 \mapsto N_1, \ldots, x_k \mapsto N_k,\\ & y_1 \mapsto \mathrm{eq}(bs'_1, M'_1), \ldots, y_\ell \mapsto \mathrm{eq}(bs'_\ell, M'_\ell)\}\end{array}$$

For any i, the probability to have y_i different from 0 in $[\![\varphi]\!]_{A_\eta}$ is lower than $1/2^\eta$. Hence the probability for one of the y_i to be different from 0 is lower than $\ell/2^\eta$. Let us denote by $P(\psi)$ the event that for all i $(1 \leq i \leq \ell)$ $y_i\psi = 0$. We have that

$$\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} \ : \ \neg P(\psi)\right] \leq \frac{\ell}{2^\eta}$$

Thus with overwhelming probability all the y_i are equal to 0. The same holds for φ'. This allows us to conclude that the advantage of $\mathcal{A}$ is negligible.

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{IND}}_{\mathcal{A}} &= \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1\right]\big| \\
&= \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge P(\psi)\right] + \\
&\quad \mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge \neg P(\psi)\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge P(\psi')\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge \neg P(\psi')\right]\big| \\
&\leq \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge P(\psi)\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge P(\psi')\right]\big| + \\
&\quad \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge \neg P(\psi)\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge \neg P(\psi')\right]\big| \\
&\leq \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge P(\psi)\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge P(\psi)\right]\big| + \\
&\quad \mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge \neg P(\psi)\right] + \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge \neg P(\psi')\right] \\
&\leq \big|\mathbb{P}\left[\psi \leftarrow [\![\varphi]\!]_{A_\eta} : \mathcal{A}(\psi) = 1 \wedge P(\psi)\right] - \\
&\quad \mathbb{P}\left[\psi' \leftarrow [\![\varphi']\!]_{A_\eta} : \mathcal{A}(\psi') = 1 \wedge P(\psi')\right]\big| + \\
&\quad \tfrac{2\ell}{2^\eta}
\end{aligned}$$

As the distributions of ψ and ψ' when $P(\psi)$ and $P(\psi')$ are true are identical, the advantage of $\mathcal{A}$ is bounded by $2\ell/2^\eta$, which is negligible. Hence, (A_η) is $\approx_E$-sound.

A.2 Proof of Proposition 2

Proposition 2 *Let (A_η) be a family of computational algebras. A_η is unconditionally $\approx_E$-ad-sound iff A_η is unconditionally $\approx_E$-sound.*

($\Rightarrow$) Suppose that (A_η) is unconditionally $\approx_E$-ad-sound. Let $\varphi = \{x_i \mapsto t_i\}_{1\leq i\leq n}$ and $\varphi' = \{x_i \mapsto u_i\}_{1\leq i\leq n}$ be two frames such that $\varphi \approx_E \varphi'$. For any computational frame f of domain $\{x_i\}_{1\leq i\leq n}$ we build an adversary $\mathcal{B}_f$ against adaptive soundness:

Adversary $\mathcal{B}_f^{\mathcal{O}_{LR}}$:
  for i **from** 1 **to** n
    $bs_i \leftarrow \mathcal{O}_{LR}(t_i, u_i)$
  if $f = \{x_i \mapsto bs_i\}_{1\leq i\leq n}$ **then return** 1
  else return 0

Because of unconditional adaptive soundness, the advan-

tage of $\mathcal{B}$ is 0. Hence, we have that

$$\begin{aligned}\mathbb{P}\left[\mathcal{B}_f^{\mathcal{O}^1_{LR,A_\eta}}=1\right] &= \mathbb{P}\left[\mathcal{B}_f^{\mathcal{O}^0_{LR,A_\eta}}=1\right]\\ \mathbb{P}\left[f'\leftarrow[\![\varphi']\!]_{A_\eta}, f=f'\right] &= \mathbb{P}\left[f'\leftarrow[\![\varphi]\!]_{A_\eta}, f=f'\right]\end{aligned}$$

We obtain that the distributions of $[\![\varphi]\!]_{A_\eta}$ and $[\![\varphi']\!]_{A_\eta}$ are equal which allows us to conclude that (A_η) is $\approx_E$-sound.

($\Leftarrow$) Suppose that (A_η) is unconditionally $\approx_E$-sound. Let $\mathcal{A}$ be an adversary against adaptive soundness. Let f be a computational frame and φ and φ' be two statically equivalent formal frames. We denote by $e(\varphi,\varphi')$ where $\mathcal{A}$ asked for frame (φ,φ') through his left-right oracle (the adversary uses one call to the oracle for each element in the domain of φ). We also denote by $e'(f)$ the event where $\mathcal{A}$ the oracle returns f to $\mathcal{A}$. As $\mathcal{A}$ is polynomially bounded, the number of possible triples (f,φ,φ') is bounded. Let S be the set of such possible triples. Let a be the advantage of $\mathcal{A}$ in the adaptive setting: we have that $a=\sum_{(f,\varphi,\varphi')\in S} p(f,\varphi,\varphi')$ where

$$\begin{aligned}p(f,\varphi,\varphi') &= \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR}}=1\middle|\begin{array}{c}e'(f)\\ e(\varphi,\varphi')\end{array}\right]\mathbb{P}\left[\begin{array}{c}e(\varphi,\varphi')\\ f\leftarrow[\![\varphi]\!]\end{array}\right]\\ &- \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR}}=1\middle|\begin{array}{c}e'(f)\\ e(\varphi,\varphi')\end{array}\right]\mathbb{P}\left[\begin{array}{c}e(\varphi,\varphi')\\ f\leftarrow[\![\varphi']\!]\end{array}\right]\end{aligned}$$

As $\mathcal{A}$ depends only on his random coins and on f, *i.e.*, $\mathcal{A}$ does not depend on the challenge bit of the oracle, we have that for any given f,

$$\mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR}}=1\middle|\begin{array}{c}e'(f)\\ e(\varphi,\varphi')\end{array}\right]=\mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR}}=1\middle|\begin{array}{c}e'(f)\\ e(\varphi,\varphi')\end{array}\right].$$

Therefore,

$$\begin{aligned}p(f,\varphi,\varphi') &= \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR}}=1\middle|\begin{array}{c}e'(f)\\ e(\varphi,\varphi')\end{array}\right]\cdot\\ &\left(\mathbb{P}\left[\begin{array}{c}e(\varphi,\varphi')\\ f\leftarrow[\![\varphi]\!]\end{array}\right]-\mathbb{P}\left[\begin{array}{c}e(\varphi,\varphi')\\ f\leftarrow[\![\varphi']\!]\end{array}\right]\right)\end{aligned}$$

Because of unconditional $\approx_E$-soundness, and as φ and φ' are statically equivalent, the distributions $[\![\varphi]\!]$ and $[\![\varphi']\!]$ are equal. Hence,

$$\mathbb{P}\left[f\leftarrow[\![\varphi]\!]_{A_\eta}\right]=\mathbb{P}\left[f\leftarrow[\![\varphi']\!]_{A_\eta}\right]$$

This implies that the advantage of $\mathcal{A}$ is 0, and we conclude that (A_η) is unconditionally $\approx_E$-ad-sound.

A.3 Proof of Proposition 3

Proposition 3 (Composition) *Let Σ_1 and Σ_2 be two disjoint signatures and S be a signature combination for Σ_1 and Σ_2. Let E_1 and E_2 be equational theories over Σ_1 and Σ_2 respectively. We consider a family of computational algebras (A^1_η) for Σ_1 and another family (A^2_η) for Σ_2 respecting S,* i.e. *$(s_2,s_1)\in\mathsf{S}$ implies that $[\![s_2]\!]_{A^2_\eta}\subseteq[\![s_1]\!]_{A^1_\eta}$.*

Let F be a set of pair of frames over $\Sigma_1\cup\Sigma_2$ and σ be a (E_1,E_2)-hybrid function for F. If $A^1_\eta\times A^2_\eta$ is $\approx_{E_1}$-ad-sound for $G=\{(\varphi,\sigma(\varphi))\mid\varphi$ occurs in $F\}$ and A^2_η is $\approx_{E_2}$-ad-sound for frames on Σ_2, then $A^1_\eta\times A^2_\eta$ is $\approx_{E_1\cup E_2}$-ad-sound for F.

Let A_η be $A^1_\eta\times A^2_\eta$ and $\mathcal{A}$ be an adversary against adaptive soundness for pairs of frames in F. We introduce a new oracle which is the hybrid implementation of oracle $\mathcal{O}^b_{LR,A_\eta}$: $\mathcal{H}^b_{LR,A_\eta}$. When called on the sequence of queries $(t^j_0,t^j_1)_{1\le j\le i}$, $\mathcal{H}^b_{LR,A_\eta}$ outputs $[\![\sigma([t^1_b,...,t^i_b])]\!]$.

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

We denote these three differences of probabilities by p_1, p_2 and p_3. We will now show that each of them corresponds to the advantage of an adversary against either $\approx_{E_1}$-ad-soundness or $\approx_{E_2}$-ad-soundness.

- $p_1=\mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR,A_\eta}}=1\right]-\mathbb{P}\left[\mathcal{A}^{\mathcal{H}^1_{LR,A_\eta}}=1\right]$

  We now construct an adversary $\mathcal{B}_1$ against $\approx_{E_1}$-ad-soundness such that his advantage is p_1.

  Adversary $\mathcal{B}_1^{\mathcal{O}_{LR}}$:
  $d\leftarrow\mathcal{A}^{S_1}$
  return d

  $\mathcal{B}_1$ uses an algorithm S_1 to answer $\mathcal{A}$'s queries which works as follows. When $\mathcal{A}$ submits a query (t^i_0,t^i_1) to his oracle $\mathcal{B}$ queries his left-right oracle with $(\sigma([t^1_1,...,t^i_1]),t_1)$. By definition of hybrid function σ, $\mathcal{B}_1$ is a legal adversary against $\approx_{E_1}$-ad-soundness for pairs of frames of the form $(\varphi,\sigma(\varphi))$. Hence the advantage p_1 of $\mathcal{B}_1$ is negligible.

- $p_2=\mathbb{P}\left[\mathcal{A}^{\mathcal{H}^1_{LR,A_\eta}}=1\right]-\mathbb{P}\left[\mathcal{A}^{\mathcal{H}^0_{LR,A_\eta}}=1\right]$

  We now construct an adversary $\mathcal{B}_2$ against $\approx_{E_1}$ soundness for frames in F such that his advantage is p_2.

Adversary $\mathcal{B}_2^{\mathcal{O}_{LR}}$:
$d \leftarrow \mathcal{A}^{S_2}$
return d

Adversary $\mathcal{B}_2$ uses an algorithm S_2 to simulate $\mathcal{A}$'s oracle and answer his queries. S_2 works as follows. Suppose $\mathcal{A}$ submits (t_0^i, t_1^i) to his oracle, *i.e.*, to S_2. If some names from Σ_1 in $\sigma([t_1^1, ..., t_1^i])$ and $\sigma([t_0^1, ..., t_0^i])$ have not been generated previously, values are randomly sampled using A_η^1 (note that $\sigma([t_0^1, ..., t_0^i])$ and $\sigma([t_1^1, ..., t_1^i])$ use exactly the same names at the same position). Let $(q_j)_j$ be the set of minimal positions in Σ_2 in $\sigma([t_1^1, ..., t_1^i])$. Then for each j, $\mathcal{B}_2$ queries his oracle $\mathcal{O}_{LR}$ on the pair $\left(\sigma([t_0^1, ..., t_0^i])|_{q_j}, \sigma([t_1^1, ..., t_1^i])|_{q_j}\right)$ resulting in a bit-string bs_j. Using the bit-strings (bs_j), values for subterms from Σ_2 and interpretation for symbols of Σ_1 in A_η^1, $\mathcal{B}_2$ constructs either $[\![\sigma([t_0^1, ..., t_0^i])]\!]$ (challenge bit 0) or $[\![\sigma([t_1^1, ..., t_1^i])]\!]$ (challenge bit 1) and returns this value to $\mathcal{A}$. By definition of hybrid function σ, this adversary is legal against $\approx_{E_2}$-ad-soundness. Hence $\mathcal{B}_2$'s advantage p_2 is negligible.

- $p_3 = \mathbb{P}\left[\mathcal{A}^{\mathcal{H}^0_{LR,A_\eta}} = 1\right] - \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR,A_\eta}} = 1\right]$

  We again build an adversary $\mathcal{B}_3$ against $\approx_{E_1}$-ad-soundness such that his advantage is p_3. $\mathcal{B}_3$ is built in a similar way as $\mathcal{B}_1$. The only difference is that $\mathcal{B}_3$ discards t_1^i and processes t_0^i when $\mathcal{A}$ queries (t_0^i, t_1^i). Hence p_3 is also negligible.

As p_1, p_2 and p_3 are negligible we conclude that $\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A},A_\eta}(\eta)$ is also negligible. Therefore $A_\eta^1 \times A_\eta^2$ is $\approx_{E_1 \cup E_2}$ -ad-sound for frames in F.

B Proofs for Section 4

B.1 Proof of Proposition 4

Proposition 4 *Let $\mathcal{SE}$ be a symmetric encryption scheme. If $\mathcal{SE}$ is* IND-CPA*, then $\mathcal{SE}$ is* IND-CPA′*. However $\mathcal{SE}$ can be* IND-CPA′ *without being* IND-CPA.

Let $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme that is IND-CPA. Let $\mathcal{A}$ be an adversary against IND-CPA' for $\mathcal{SE}$. We build an adversary $\mathcal{B}$ against IND-CPA as follows: when $\mathcal{A}$ queries his oracle with a list of pairs $(bs_0^i, bs_1^i)_{1\leq i\leq n}$, $\mathcal{B}$ makes n queries to his left-right encryption oracle with (bs_0^i, bs_1^i). The oracle returns bs_i for each of the n queries and $\mathcal{B}$ answers $\mathcal{A}$'s query with the list $(bs_i)_{1\leq i\leq n}$. The execution of adversaries $\mathcal{B}$ and $\mathcal{A}$ are exactly the same hence they have the same advantage. As the advantage of B is negligible by hypothesis, the advantage of $\mathcal{A}$ is negligible too. Hence, $\mathcal{SE}$ is IND-CPA'.

The converse is not true in general, as shown by the following counter-example. Let us suppose the existence of an IND-CPA encryption scheme $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$. We build an encryption scheme $\mathcal{SE}' = (\mathcal{KG}', \mathcal{E}', \mathcal{D}')$ that is IND-CPA' but not IND-CPA.

- $\mathcal{KG}'$ outputs a pair (k, bs) where k is a key generated using $\mathcal{KG}$ and bs is a randomly sampled bit-string in $[0,1]^\eta$ (with uniform probability).
- $\mathcal{E}'(m, (k, bs))$ returns 0 if m is equal to bs, $1 \cdot bs \cdot \mathcal{E}(m, k)$ otherwise.
- $\mathcal{D}'(m, (k, bs))$ returns bs if m equals 0, otherwise, if $m = 1 \cdot bs \cdot m'$, it returns $\mathcal{D}(m', k)$.

There is a trivial attack against IND-CPA for $\mathcal{SE}'$ which yields an adversary of advantage 1: the adversary gives $(0,0)$ to his left-right encryption oracle which returns $1 \cdot bs \cdot \mathcal{E}(0,k)$ (note that $0 \neq bs$ for any $\eta > 1$). The adversary gives $(0, bs)$ to his left-right encryption oracle. If the oracle returns 0, he outputs 1, otherwise he outputs 0. It is straightforward to see that the adversary outputs the challenge bit. Hence, $\mathcal{SE}'$ is not IND-CPA.

We now show that $\mathcal{SE}'$ is IND-CPA′. Let $\mathcal{A}$ be an adversary against IND-CPA' for $\mathcal{SE}'$. We build an adversary $\mathcal{B}$ against IND-CPA for $\mathcal{SE}$ using $\mathcal{A}$. $\mathcal{B}$ starts by randomly sampling a bit-string bs in $[0,1]^\eta$ before executing $\mathcal{A}$. When $\mathcal{A}$ queries his oracle with a list of pairs $(bs_0^i, bs_1^i)_{1\leq i\leq n}$, if one of the bs_b^i is equal to bs then $\mathcal{B}$ halts and outputs 0. Else $\mathcal{B}$ makes n queries (bs_0^i, bs_1^i) to his left-right encryption oracle. The oracle returns bs_i for each of the n queries and $\mathcal{B}$ answers $\mathcal{A}$'s query with the list $(1 \cdot bs \cdot bs_i)_{1\leq i\leq n}$. Except for the event E where one of the bs_b^i is equal to bs, $\mathcal{B}$ correctly simulates the oracle and the execution of the adversaries $\mathcal{B}$ and $\mathcal{A}$ are exactly the same. Hence the advantage of $\mathcal{A}$ is given by:

$$\begin{aligned}
\mathrm{Adv}^{cpa'}_{\mathcal{SE}',\mathcal{A}} &= \left|\mathbb{P}\left[\mathcal{A}^{LR^1_{\mathcal{S}\varepsilon}} = 1 \wedge E\right] + \mathbb{P}\left[\mathcal{A}^{LR^1_{\mathcal{S}\varepsilon}} = 1 \wedge \neg E\right] - \right. \\
&\qquad \left. \mathbb{P}\left[\mathcal{A}^{LR^0_{\mathcal{S}\varepsilon}} = 1 \wedge E\right] - \mathbb{P}\left[\mathcal{A}^{LR^0_{\mathcal{S}\varepsilon}} = 1 \wedge \neg E\right]\right| \\
\mathrm{Adv}^{cpa'}_{\mathcal{SE}',\mathcal{A}} &\leq \left|\mathbb{P}\left[\mathcal{A}^{LR^1_{\mathcal{S}\varepsilon}} = 1 \wedge E\right] - \mathbb{P}\left[\mathcal{A}^{LR^0_{\mathcal{S}\varepsilon}} = 1 \wedge E\right]\right| + \\
&\qquad \left|\mathbb{P}\left[\mathcal{A}^{LR^1_{\mathcal{S}\varepsilon}} = 1 \wedge \neg E\right] - \mathbb{P}\left[\mathcal{A}^{LR^0_{\mathcal{S}\varepsilon}} = 1 \wedge \neg E\right]\right| \\
\mathrm{Adv}^{cpa'}_{\mathcal{SE}',\mathcal{A}} &\leq 2\mathbb{P}[E] + \mathrm{Adv}^{cpa}_{\mathcal{SE},\mathcal{B}}
\end{aligned}$$

As n is polynomially bounded in η, the probability of E, which is bounded by $1 - (1 - 1/2^\eta)^{2n}$, is negligible[2] in η. Hence the advantage of $\mathcal{A}$ is negligible and $\mathcal{SE}$ is IND-CPA'.

[2] Indeed we have that $1 - (1 - 1/2^\eta)^{2n} \leq 2n/2^\eta$

B.2 Proof sketch of Proposition 5

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

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

We prove a more general result which will be useful when combining symmetric encryption and modular exponentiation (in Appendix B.5): IND-CPA implies adaptive soundness of symbolic encryption even when symbols from E_{sym} are combined with other symbols.

We use the notion of patterns from [4]:

$$\begin{array}{rcll} \mathsf{pat}\,((t_0,t_1),K) & = & (\mathsf{pat}\,(t_0,K),\mathsf{pat}\,(t_1,K)) & \\ \mathsf{pat}\,(\{t\}_k,K) & = & \{\mathsf{pat}\,(t,K)\}_k & \text{if } k\in K \\ \mathsf{pat}\,(\{t\}_k,K) & = & \{\Box\}_k & \text{if } k\notin K \\ \mathsf{pat}\,(t,K) & = & t & \text{else} \end{array}$$

Two sequences of terms $(t_0^1,\ldots,t_0^\ell)$ and $(t_1^1,\ldots,t_1^\ell)$ are said to be *p-equivalent*, denoted $(t_0^1,\ldots,t_0^\ell)\equiv(t_1^1,\ldots,t_1^\ell)$ if $\mathsf{pat}\left(t_0^i,K_0\right)=\mathsf{pat}\left(t_1^i,K_1\right)$ $(1\le i\le\ell)$ where K_0, respectively K_1, is the set of keys deducible from the frame $\{x_1\mapsto t_0^1,\ldots,x_\ell\mapsto t_0^\ell\}$, respectively $\{x_1\mapsto t_1^1,\ldots,x_\ell\mapsto t_1^\ell\}$.

Let $\prec$ be a total order among keys, an adversary against adaptive soundness is *p-legal* for $\prec$ if he only makes queries $(t_0^i,t_1^i)_{1\le i\le\ell}$ such that:

- $(t_0^1,\ldots,t_0^\ell)\equiv(t_1^1,\ldots,t_1^\ell)$;
- both frames $\{x_1\mapsto t_0^1,\ldots,x_\ell\mapsto t_0^\ell\}$ and $\{x_1\mapsto t_1^1,\ldots,x_\ell\mapsto t_1^\ell\}$ are acyclic for $\prec$ and the only symbols from Σ_{sym} are enc, pair, 0 and 1;
- if k is used as plaintext in t_b^i, then k cannot be used at a key position in t_b^j for $j<i$.

Lemma 1 *Let Σ_2 be a disjoint signature from Σ_{sym} and* S *be a signature combination for Σ_{sym} and Σ_2. Let (A_η^1) be a family of computational algebras for Σ_{sym} based on an* IND-CPA *symmetric encryption scheme and (A_η^2) be a family of computational algebras for Σ_2. Let $\prec$ be a total order among keys, then $A_\eta^1\times A_\eta^2$ is $\approx_{E_{\mathsf{sym}}}$-ad-sound for p-legal adversaries for $\prec$ using terms on the $(\Sigma_{\mathrm{sym}},\Sigma_2)_S$ layered signature $\Sigma_{\mathrm{sym}}\cup\Sigma_2$.*

Proof: The proof of this result is close to the one of the main result from [28] (which itself uses the same hybrid argument as the proof of [11]). Let $\mathcal{SE}=(\mathcal{KG},\mathcal{E},\mathcal{D})$ be the symmetric encryption scheme used by A_η^1. Let $\mathcal{A}$ be an adversary against E_{sym}-ad-soundness and P be a polynomial bounding the execution time of $\mathcal{A}$. Without loss of generality queries made by $\mathcal{A}$ use keys k_1 to $k_{P(\eta)}$ such that $k_i\prec k_j$ iff $i<j$. We build 2 adversaries $\mathcal{B}_b$ against IND-CPA (where b ranges over $[0,1]$). The idea is that each $\mathcal{B}_b$ randomly samples a value i between 1 and $P(\eta)$, uses his IND-CPA challenge key for k_i and simulates the other keys.

Adversary $\mathcal{B}_b$ randomly samples an integer i between 1 and $P(\eta)$. $\mathcal{B}_b$ also generates a value for each key k_j using the key generation algorithm $\mathcal{KG}$. A value is even sampled for key k_i which will be used if $\mathcal{A}$ asks for revelation of k_i, (*i.e.*, if k_i becomes deducible from requests issued by $\mathcal{A}$). $\mathcal{B}_b$ then executes $\mathcal{A}$ and uses an algorithm S to answer queries made by $\mathcal{A}$.

Adversary $\mathcal{B}_b^{LR_{\mathcal{SE}}}$:
$i\leftarrow[1,P(\eta)]$
for j **from** 1 **to** $P(\eta)$ **do** $k_j\leftarrow\mathcal{KG}(\eta)$
$d\leftarrow\mathcal{A}^S$
return d

S uses the values k_j generated by $\mathcal{B}_b$ and also generates and stores values for other names used by $\mathcal{A}$. Let $(t_0^l,t_1^l)_{1\le l\le\ell}$ be the previous p-equivalent queries made by $\mathcal{A}$ (including the current query (t_0^ℓ,t_1^ℓ)), the algorithm S works recursively on pairs of terms (t_0,t_1):

- If t_0 is a pair (t_0',t_0'') then as the sequences of queries are p-equivalent, t_1 is a pair (t_1',t_1'') and $(t_0',t_0'')\equiv(t_1',t_1'')$. S is recursively applied on (t_0',t_1') yielding bs' and on (t_0'',t_1'') yielding bs''. S returns the concatenation of bs' and bs''.
- If t_0 is an encryption $\{t_0'\}_{k_j}$ and $\{x_l\mapsto t_0^l\}_{1\le l\le\ell}\vdash k_j$ then t_1 is an encryption $\{t_1'\}_{k_j}$ and $t_0'\equiv t_1'$. Algorithm S is applied recursively on (t_0',t_1') yielding bs'. S returns $\mathcal{E}(bs',k_j)$.
- If t_0 is an encryption $\{t_0'\}_{k_j}$ and $\{x_l\mapsto t_0^l\}_{1\le l\le\ell}\not\vdash k_j$, then t_1 is an encryption $\{t_1'\}_{k_j}$. S is applied recursively on (t_b',t_b') yielding a bit-string bs_b.

$$S\text{ returns }\begin{cases}\mathcal{E}(bs_b,k_j) & \text{if } j<i\\ LR_{\mathcal{SE}}(0,bs_b) & \text{if } j=i\\ \mathcal{E}(0,k_j) & \text{if } j>i\end{cases}$$

Note that in the case where $j=i$, $\mathcal{B}_b$ uses his left-right encryption oracle so S outputs $LR(0,bs_b)$ which is the encryption of either bs_b or 0 under the challenge key.

- If t_0 is a key name k_j then t_1 is the same key k_j and S outputs the generated value k_j.

- If t_0 is a constant 0 or 1 then t_1 is the same constant and S outputs either $[\![0]\!]_{A^1_\eta}$ or $[\![1]\!]_{A^1_\eta}$.
- If t_0 is a term using Σ_2 then t_1 is equal to t_0 and S outputs $[\![t_0]\!]_{A^2_\eta}$.

An important point to note is that the adversary is p-legal. If $\mathcal{B}_b$ uses his left-right encryption oracle then k_i is used as key in a request t^b_j and is not deducible from previous requests. Therefore k_i is not deducible from further requests (as it cannot be used as plaintext anymore). Moreover occurrences of k_i as plaintexts were always encrypted using a non-deducible key k_j (for $j > i$ because of acyclicity). Hence the algorithm S does not use the value of key k_i (as 0 is encrypted instead). Therefore the algorithm S is coherent. We sum up in the following array how the algorithm S proceeds with encryption using a non-deducible key k_j (0 means that 0 is encrypted, bs_b indicates that the part corresponding to t_b is encrypted).

	k_1	$\dots$	k_{i-1}	k_i	k_{i+1}	$\dots$	$k_{P(\eta)}$
$\mathcal{B}_b$	bs_b	$\dots$	bs_b	$bs_b/0$	0	$\dots$	0

For key k_i, bs_b is used if the challenge bit of $\mathcal{B}_b$ is 1, else 0 is used. When the challenge bit of $\mathcal{B}_0$ is 1 and i equals $P(\eta)$, $\mathcal{A}$ is executed normally with challenge bit 0 and hence

$$\mathbb{P}\left[i = P(\eta) \ : \ \mathcal{B}_0^{LR^1_{S\mathcal{E}}} = 1\right] = \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR}} = 1\right]$$

When the challenge bit of $\mathcal{B}_b$ is 1, algorithm S encrypts 0 for keys k_{i+1} to $k_{P(\eta)}$. The same thing happens when the challenge bit of $\mathcal{B}_b$ is 0 and the value $i+1$ is generated instead of i. Hence

$$\mathbb{P}\left[\ i = x \ : \ B_b^{LR^1_{S\mathcal{E}}} = 1\right] = \mathbb{P}\left[i = x+1 \ : \ \mathcal{B}_b^{LR^0_{S\mathcal{E}}} = 1\right]$$

When the challenge bit of $\mathcal{B}_0$ is 0 and i equals 1, when $\mathcal{A}$ performs queries $(t^l_0, t^l_1)_{1\le l\le \ell}$, he receives interpretations for $\mathsf{pat}\left(t^l_0, K_0\right)$ (where $\square$ is interpreted with 0 and K_0 is the set of keys deducible from $\{x_l \mapsto t^l_0\}_{1\le l\le \ell}$). When the challenge bit of $\mathcal{B}_1$ is 0 and i equals 1, when $\mathcal{A}$ performs queries $(t^l_0, t^l_1)_{1\le l\le \ell}$ to his left-right oracle, he receives interpretations for $\mathsf{pat}\left(t^l_1, K_1\right)$ which is equal to $\mathsf{pat}\left(t^l_0, K_0\right)$. Therefore, we have that

$$\mathbb{P}\left[i = 1 \ : \ \mathcal{B}_0^{LR^0_{S\mathcal{E}}} = 1\right] = \mathbb{P}\left[i = 1 \ : \ \mathcal{B}_1^{LR^0_{S\mathcal{E}}} = 1\right]$$

Finally when the challenge bit of $\mathcal{B}_1$ is 1 and i equals $P(\eta)$, $\mathcal{A}$ is executed normally with challenge bit 1. Hence we have that

$$\mathbb{P}\left[i = P(\eta) \ : \ \mathcal{B}_1^{LR^1_{S\mathcal{E}}} = 1\right] = \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR}} = 1\right]$$

To shorten notations, we write $\mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{x,b} = 1\right]$ for probability $\mathbb{P}\left[i = x \ : \ \mathcal{B}_b^{LR^1_{S\mathcal{E}}} = 1\right]$ The advantage of $\mathcal{A}$ is given by:

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A}}(\eta) &= \left|\mathbb{P}\left[\mathcal{A}^{\mathcal{O}^1_{LR}} = 1\right] - \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^0_{LR}} = 1\right]\right| \\
&= \left|\mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{P(\eta),1} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{P(\eta),0} = 1\right]\right| \\
&= \left|\mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{P(\eta),1} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{P(\eta),1} = 1\right] + \right. \\
&\qquad \vdots \\
&\qquad \mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{1,1} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{1,1} = 1\right] + \\
&\qquad \mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{1,1} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{1,0} = 1\right] + \\
&\qquad \mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{1,0} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{1,0} = 1\right] + \\
&\qquad \vdots \\
&\qquad \left.\mathbb{P}\left[\mathcal{B}^{LR^0_{S\mathcal{E}}}_{P(\eta),0} = 1\right] - \mathbb{P}\left[\mathcal{B}^{LR^1_{S\mathcal{E}}}_{P(\eta),0} = 1\right]\right| \\
&\le P(\eta)\,\mathrm{Adv}^{cpa}_{\mathcal{SE},\mathcal{B}_1}(\eta) + P(\eta)\,\mathrm{Adv}^{cpa}_{\mathcal{SE},\mathcal{B}_0}(\eta)
\end{aligned}$$

As $\mathcal{SE}$ is IND-CPA the advantage of each $\mathcal{B}_b$ is negligible so the advantage of $\mathcal{A}$ is also negligible and we obtain adaptive soundness for p-legal adversaries. ■

The first implication of Proposition 5 is a consequence of the previous proposition by using an empty signature Σ_2 and the following lemma which states that p-equivalent terms correspond to statically equivalent frames.

Lemma 2 *Let $\varphi = \{x_i \mapsto t_i\}_{1\le i\le n}$ and $\varphi' = \{x_i \mapsto t'_i\}_{1\le i\le n}$ be two well-formed frames.*

$$\varphi \approx_{E_{\mathsf{sym}}} \varphi' \Rightarrow (t_1, \dots, t_n) \equiv (t'_1\rho, \dots, t'_n\rho)$$

where ρ is some bijective renaming of keys.

Proof: We suppose that φ and φ' contain only one term, *i.e*, $\varphi = \{x \mapsto t\}$ and $\varphi' = \{x \mapsto t'\}$. This hypothesis is not a loss of generality in the theory E_{sym} as sequences of terms can be grouped into a single term using the pair operator.

Let K and K' be the sets of deducible keys from respectively φ and φ', let u be $\mathsf{pat}\,(t, K)$ and u' be $\mathsf{pat}\,(t', K')$. We suppose that $u \neq u'\rho$ for any bijective renaming ρ and show that $\varphi \approx_{E_{\mathsf{sym}}} \varphi'$.

Let p be the minimal position, such that

- $p \in pos(u) \cap pos(u')$;
- $root(u|_p) \neq root(u'|_p)$;
- either $root(u|_p)$ or $root(u'|_p)$ is not a name.

We have to consider two cases according to whether such a p exists.

If such a position p exists and $root(u|_p) \neq \Box$, we denote by M_p the term such that $M_p\varphi =_{E_{\mathsf{sym}}} u|_p$. Note that as p is minimal, p cannot be a key position. It directly follows from the definition of patterns that such a term M_p exists. We have one of the following cases:

- $root(u|_p) = \mathrm{pair}$
  Then we have that $\mathrm{tpair}(M_p\varphi) =_{E_{\mathsf{sym}}} 1$ while $\mathrm{tpair}(M_p\varphi') \neq_{E_{\mathsf{sym}}} 1$.
- $root(u|_p) = \mathrm{enc}$
  Then we have that $\mathrm{tenc}(M_p\varphi) =_{E_{\mathsf{sym}}} 1$ while $\mathrm{tenc}(M_p\varphi') \neq_{E_{\mathsf{sym}}} 1$.
- $root(u|_p) = \Box$
  Then we have that $root(u|_{p'}) = \mathrm{enc}$ where $p = p' \cdot 1$. Moreover, $root(u|_{p'}) = root(u'|_{p'})$, as p is minimal, and $root(u'|_p) \neq \Box$. Hence, by definition of patterns the key at position $p' \cdot 2$ and the term at position p' are deducible from φ'. We denote by M_k and $M_{p'}$ the terms such that such that $M_k\varphi' =_{E_{\mathsf{sym}}} u'|_{p'\cdot 2}$ and $M_{p'}\varphi' =_{E_{\mathsf{sym}}} u'|_{p'}$. Then, we have that $\mathrm{samekey}(M_{p'}, \mathrm{enc}(0, M_k))\varphi' =_{E_{\mathsf{sym}}} 1$, while this test does not hold under φ.
- $root(u|_p) = 0$
  Then we have that $M_p\varphi =_{E_{\mathsf{sym}}} 0$ while $M_p\varphi' \neq_{E_{\mathsf{sym}}} 0$.
- $root(u|_p) = 1$
  Then we have that $M_p\varphi =_{E_{\mathsf{sym}}} 1$ while $M_p\varphi' \neq_{E_{\mathsf{sym}}} 1$.
- $root(u|_p)$ is a name
  By definition of p, $root(u'|_p)$ is not a name. Hence this case is symmetric to one of the above cases.

If no such position p exists, then the patterns only differ by names. As, there exists no bijective renaming there must be a name which is used twice in one of the frames, but two different names are used at the same positions in the other frame. Let p_1 and p_2 be the positions of these two names. We consider the following cases.

- p_1 and p_2 are both plain text positions
  From the definition of patterns it follows that they are both deducible using terms which we denote by M_{p_1} and M_{p_2} respectively. The test $M_{p_1} = M_{p_2}$ distinguishes φ and φ'.
- p_1 and p_2 are both key positions
  Let $p_1 = p_1' \cdot 2$ and $p_2 = p_2' \cdot 2$. Then, the test $\mathrm{samekey}(M_{p_1'}, M_{p_2'}) =_{E_{\mathsf{sym}}} 1$ distinguishes distinguishes φ and φ', where $M_{p_1'}$, respectively $M_{p_2'}$, denotes the term such that $M_{p_1'}\varphi =_{E_{\mathsf{sym}}} t|_{p_1'}$, respectively $M_{p_2'}\varphi =_{E_{\mathsf{sym}}} t|_{p_2'}$.
- p_1 is a plain text position and p_2 is a key position
  We use the same notations as in the two previous cases. The test $\mathrm{samekey}(M_{p_2'}, \mathrm{enc}(0, M_{p_1})) =_{E_{\mathsf{sym}}} 1$ distinguishes φ and φ'.
- p_1 is a key position and p_2 is a plain text position
  This case is symmetric to the previous one.

■

Note that the converse of Lemma 2 is not true, *i.e.*, p-equivalence does not imply static equivalence. This is due to the fact that randomness of the encryption is kept implicit in the formal model. Indeed we have that $(\{0\}_k, \{0\}_k) \equiv (\{0\}_k, \{1\}_k)$ while $\{x_1 \mapsto \{0\}_k, x_2 \mapsto \{0\}_k\} \not\approx_{E_{\mathsf{sym}}} \{x_1 \mapsto \{0\}_k, x_2 \mapsto \{1\}_k\}$ (the test $x_1 = x_2$ distinguishes both frames). A more precise result could be established by modelling the randomness explicitly in both the signature and the patterns.

Using Lemma 2, we obtain that any well-formed adversary is also p-legal. Using Lemma 1 where Σ_2 is the empty signature, we conclude that (A_η) is $\approx_{E_{\mathsf{sym}}}$-ad-sound if $\mathcal{SE}$ is IND-CPA.

We now prove that the converse of the first implication of Proposition 5 is false in general. For this purpose we build an encryption scheme $\mathcal{SE}' = (\mathcal{KG}', \mathcal{E}', \mathcal{D}')$ which guarantees adaptive soundness but is not IND-CPA. Let $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ be an IND-CPA symmetric encryption scheme. We assume wlog that $\mathcal{KG}$ never outputs 1 and for any s $\mathcal{E}'(s, k)$ is different from 1. $\mathcal{SE}'$ is defined as follows.

- $\mathcal{KG}'$ executes algorithm $\mathcal{KG}$ and returns its output.
- $\mathcal{E}'(s, k)$ returns 1 if $s = 1$, $\mathcal{E}(s, k)$ otherwise.
- $\mathcal{D}'(c, k)$ returns 1 if $c = 1$, $\mathcal{D}(c, k)$ otherwise.

It is obvious that $\mathcal{SE}$ is neither IND-CPA nor IND-CPA': an adversary can query his oracle with the pair $(0, 1)$ to recover challenge bit with probability 1.

Let (A_η) and (A'_η) be two families of computational algebras using respectively $\mathcal{SE}$ and $\mathcal{SE}'$. As $\mathcal{SE}$ is IND-CPA (A_η) is $\approx_{E_{\mathsf{sym}}}$-ad-sound. Moreover consider an adversary $\mathcal{A}$ against adaptive soundness for (A'_η). The oracle of $\mathcal{A}$ cannot be queried to encrypt 1 (this oracle only encrypts ciphertexts, pairs or keys which are always different from 1 for $\eta > 1$). Hence the distributions output by the oracle for (A'_η) and (A_η) are the same and we conclude that (A'_η) is $\approx_{E_{\mathsf{sym}}}$-ad-sound.

The proof for the second part of Proposition 5 can be adapted from the proof of the first part in a straightforward way. The only thing to note is that as we consider soundness, we want to prove computational indistinguishability of the concrete semantics of two *fixed* frames φ and φ'. So the left-right encryption oracle of adversary $\mathcal{B}_{i,b}$ does not need to be called in an adaptive way and IND-CPA' is sufficient for this proof.

B.3 Proof of Proposition 7

Proposition 7 *We have that*

$$\{x_1 \mapsto \exp(p_1), ..., x_n \mapsto \exp(p_n)\} \approx_{E_{\mathsf{DH}}} \{x_1 \mapsto \exp(q_1), ..., x_n \mapsto \exp(q_n)\}$$

iff for any sequence of integer $a_0, a_1, \dots a_n$

$$a_0 + \sum_{i=1}^{n} a_i p_i = 0 \Leftrightarrow a_0 + \sum_{i=1}^{n} a_i q_i = 0$$

$\Rightarrow$: Suppose that there exists a sequence of integers $a_0, a_1, \dots a_n$ such that:

$$a_0 + \sum_{i=1}^{n} a_i p_i = 0 \text{ and } a_0 + \sum_{i=1}^{n} a_i q_i \neq 0$$

We build two terms M and N according to the values of the a_i: if a_0 is positive, M_0 is defined as $\exp(a_0)$ (where a_0 is itself $1_R + ... + 1_R$) and N_0 is defined as $\exp(0_R)$. Else if a_0 is negative, M_0 is defined as $\exp(0_R)$ and N_0 is defined as $\exp(-a_0)$. For i between 1 and n, if a_i is positive, then $M_i = \underbrace{x_i * \ldots * x_i}_{a_i \ times}$ and $N_i = \exp(0_R)$. Else if a_i is negative, $M_i = \exp(0_R)$ and $N_i = \underbrace{x_i * \ldots * x_i}_{-a_i \ times}$. Terms M and N are respectively given by $M = M_0 * M_1 * ... * M_n$ and $N = N_0 * N_1 * ... * N_n$. Then we have that:

$$M\{x_i \mapsto p_i\}_i =_{E_{\mathsf{DH}}} N\{x_i \mapsto p_i\}_i$$
$$M\{x_i \mapsto q_i\}_i \neq_{E_{\mathsf{DH}}} N\{x_i \mapsto q_i\}_i$$

Hence, frames $\{x_i \mapsto \exp(p_i)\}_i$ and $\{x_i \mapsto \exp(q_i)\}_i$ are not statically equivalent for E_{DH}.

$\Leftarrow$: Suppose that both frames are not statically equivalent. Hence we have that there exists M and N such that

$$M\{x_i \mapsto p_i\}_i =_{E_{\mathsf{DH}}} N\{x_i \mapsto p_i\}_i$$
$$M\{x_i \mapsto q_i\}_i \neq_{E_{\mathsf{DH}}} N\{x_i \mapsto q_i\}_i$$

(or the converse situation). We only need to consider tests where M and N are of sort G as all elements in the frame are of sort G. Hence, M and N are of the form $\prod_{j=1}^{m_M} x_{\alpha_j} * \prod_{k}^{\ell_M} \exp(b_k)$, respectively $\prod_{j=1}^{m_N} x_{\beta_j} * \prod_{k=1}^{\ell_N} \exp(c_k)$ (up to associativity and commutativity of $*$). We define

$$a_0 = \sum_{k=1}^{\ell_M} b_k - \sum_{k=1}^{\ell_M} c_k$$

and $a_i = |\{j \mid \alpha_j = i\}| - |\{j \mid \beta_j = i\}|$ for $1 \leq i \leq n$. Then we have that

$$a_0 + \sum_{i=1}^{n} a_i p_i = 0 \text{ and } a_0 + \sum_{i=1}^{n} a_i q_i \neq 0.$$

B.4 Proof of Proposition 8

Proposition 8 *Let* (A_η) *be a family of computational algebras.* (A_η) *is* $\approx_{E_{\mathsf{DH}}}$*-sound iff* (A_η) *satisfies the* DDH *assumption.* (A_η) *is* $\approx_{E_{\mathsf{DH}}}$*-ad-sound iff* (A_η) *satisfies the* DDH *assumption.*

In order to prove this result, we borrow the proof technique from [15] and introduce an adaptive version of the main result of this paper. This new extension of DDH is called the *Dynamic Decisional Diffie-Hellman* (3DH) assumption and encompasses all the previous extensions of DDH that are equivalent to DDH. The next subsection introduces 3DH and proves it equivalent to DDH. We finish the proof of the proposition in the following subsection.

B.4.1 Dynamic Decisional Diffie-Hellman

Let X be a countable set of variables. A *monomial* is a product of *distinct* variables. Hence it can be represented by a finite subset of X. The order of a monomial is the number of its variables. We consider *polynomials* that are linear combination of monomials.

Introducing 3DH The 3DH assumption considers an adversary that can be confronted to two different games. As usual in decisional assumptions, the adversary has to guess against which game he is playing. For that purpose he has access to two oracles. These oracles use randomly sampled values (in $\mathbb{Z}_q$) for a finite subset $(x_1, \dots, x_\alpha)$ of X. Let n be a positive integer. Let m_1 to m_n be n different monomials. Then a vector $\vec{m}$ of size n is computed, this vector contains the integers values for the different monomials.

The first oracle behaves in the same way in the two different games, it takes as argument a polynomial P over $(x_i)_i$ (represented by a vector $\vec{v}$ of coefficients in $\mathbb{Z}_q$) and returns $g^{P(x_1,\dots,x_\alpha)}$ ($g^{\vec{v}.\vec{m}}$). In the first game, the second oracle behaves in the exact same way as the first oracle. However, in the second game, this second oracle still takes as argument a polynomial P but it returns g^r for some randomly sampled r in $\mathbb{Z}_q$.

Without restrictions, this game would be easy to win: an adversary can first submit a polynomial to the first oracle then submit it to the second oracle. If the two results are the same, the adversary knows with high probability that he is in the first game. Otherwise, the adversary knows for sure that he is in the second game. Hence we add a restriction over possible queries: let V_1 be the set of vectors asked to the first oracle and V_2 be the set of vector submitted to the second oracle, then $V_1 \cup V_2$ has to be a family of linearly independent vectors, *i.e.*, requests are independent one from another.

It is easy to see that this new 3DH assumption implies the DDH assumption: an adversary can ask for g^{x_1} and g^{x_2} to its first oracle, then he queries the second oracle with polynomial $x_1.x_2$ and receives either $g^{x_1.x_2}$ or g^r. Finally the adversary has to deduce in which situation he is.

We write $\mathcal{A}/\lambda x.F(x), \lambda x.F'(x)$ to denote an adversary $\mathcal{A}$ that can access two oracles, The first oracle takes as input x and returns $F(x)$ whereas the second one returns $F'(x)$. Game $G^1_{\mathcal{A}}(m_1, \ldots, m_n)$ and game $G^0_{\mathcal{A}}(m_1, \ldots, m_n)$ are formally defined by:

Game $G^1_{\mathcal{A}}(m_1, \ldots, m_n)$:
for i **from** 1 **to** α
$x_i \leftarrow \mathbb{Z}_q$
$\vec{m} \leftarrow (m_1[(x_i)_i], \ldots, m_n[(x_i)_i])$
$d \leftarrow \mathcal{A}/\lambda\vec{v}.g^{\vec{v}.\vec{m}}$,
$\lambda\vec{v}.g^{\vec{v}.\vec{m}}$
return d

Game $G^0_{\mathcal{A}}(m_1, \ldots, m_n)$:
for i **from** 1 **to** α
$x_i \leftarrow \mathbb{Z}_q$
$\vec{m} \leftarrow (m_1[(x_i)_i], \ldots, m_n[(x_i)_i])$
$d \leftarrow \mathcal{A}/\lambda\vec{v}.g^{\vec{v}.\vec{m}}$,
$\lambda\vec{v}.g^r$ where $r \leftarrow \mathbb{Z}_q$
return d

The first oracle is called the *real oracle* whereas the second is called the *RR oracle* The advantage of an adversary $\mathcal{A}$ is given by:

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}}^{G(m_1,\ldots,m_n)} &= \mathbb{P}\left[G^1_{\mathcal{A}}(m_1, \ldots, m_n) = 1\right] \\ &\quad -\mathbb{P}\left[G^0_{\mathcal{A}}(m_1, \ldots, m_n) = 1\right]\end{aligned}$$

Equivalence with DDH. The main result concerning 3DH is that it is equivalent to DDH. In a first proposition, we detail how an advantage against G can be reduced into multiple advantages against DDH.

Proposition 12 *Let Γ be a finite list of monomials and let α be the number of different variables that are used in Γ. We suppose that if $m_1.m_2$ appears in Γ, then m_1 and m_2 also appear in Γ. If $\mathcal{A}$ is an adversary against $G(\Gamma)$, then there exists an adversary $\mathcal{B}$ against DDH such that:*

$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma)} = 2\left(|\Gamma| - \alpha\right)\mathrm{Adv}_{\mathcal{B}}^{\mathsf{DDH}}$$

Proof:

This proof uses four intermediate lemmas. The first lemma tells us that in a finite field (with q elements), the number of solutions for a linear system of n independent equations among α variables is $q^{\alpha-n}$.

Lemma 3 *Let q be a prime. Let us consider a linear system of n equations implying α variables in $\mathbb{Z}_q$.*

$$\forall i \in [1, n], \sum_{j=1}^{\alpha} v_{i,j}.x_j = a_i$$

If vectors $\vec{v}_i = (v_{i,1}, \ldots, v_{i,\alpha})$ are linearly independent, then the number of solutions of the system is given by:

$$\left|\left\{\vec{x} \in \mathbb{Z}_q^{\alpha} | \forall i \in [1, n], \sum_{j=1}^{\alpha} v_{i,j}.x_j = a_i\right\}\right| = q^{\alpha-n}$$

Proof: This proof can be done using an induction on the number α of variables.

1. If α equals 1, then linear independence implies that there is at most one equation $v.x = a$ Let us first consider that there are no equations, then the number of solution is q:
$$\left|\left\{x \in \mathbb{Z}_q\right\}\right| = q$$

2. If α equals 1 and there is one equation. As $\mathbb{Z}_q$ is a field, this equation has exactly one solution $x = v^{-1}.a$. Hence,
$$\left|\left\{x \in \mathbb{Z}_q | v.x = a\right\}\right| = 1$$

3. In the case of $\alpha + 1$ variables, let us consider the first equation
$$\sum_{j=1}^{\alpha} v_{1,j}.x_j = a_1$$
Linear independence implies that there exists a coefficient k such that $v_{1,k}$ is different from 0. Then x_k is given by:
$$x_k = v_{1,k}^{-1}.\left(a - \sum_{j \neq k} v_{1,j}.x_j\right)$$
This substitution is used to transform the $n-1$ other equations into equations where x_k does not appear any more. The new system of equation is:
$$\forall i \in [2, n], \sum_{j \neq k}(v_{i,j} - v_{1,k}^{-1}.v_{1,j}).x_j = a_i + v_{1,k}^{-1}.a_1$$
Linear independence of the first system implies linear independence for this new system. As this system only involves $\alpha - 1$ equations, the induction hypothesis holds hence:
$$\begin{aligned}&\left|\left\{x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{\alpha} \in \mathbb{Z}_q^{\alpha-1} | \forall i \in [2, n],\right.\right. \\ &\left.\left.\sum_{j \neq k}(v_{i,j} - v_{1,k}^{-1}.v_{1,j}).x_j = a_i + v_{1,k}^{-1}.a_1\right\}\right| \\ &= q^{(\alpha-1)-(n-1)}\end{aligned}$$

Then for each solution of the new system, there is only one possibility for x_k. Hence the number of solutions of both systems are the same thus:

$$\left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha | \forall i \in [1,n], \sum_{j=1}^{\alpha} v_{i,j}.x_j = a_i\right\}\right| = q^{\alpha-n}$$

■

Using the first lemma, it is possible to relate the number of solutions of two closely linked systems.

Lemma 4 *Let q be a prime. Let us consider n (resp. m) vectors of $\mathbb{Z}_q^\alpha$ denoted by $\vec{v}_i$ (resp. $\vec{w}_j$) such that all these vectors are linearly independent. Then*

$$\begin{aligned} & \left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha \middle| \begin{array}{c} \forall i \in [1,n],\ \vec{v}_i.\vec{x} = a_i \\ \forall j \in [1,m],\ \vec{w}_j.\vec{x} = b_j \end{array}\right\}\right|.q^m \\ = & \left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha, \vec{r} \in \mathbb{Z}_q^m, \middle| \begin{array}{c} \forall i \in [1,n],\ \vec{v}_i.\vec{x} = a_i \\ \forall j \in [1,m],\ \vec{r}_j = b_j \end{array}\right\}\right| \end{aligned}$$

Proof: Linear independence allows us to apply Lemma 3 twice: First,

$$\left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha \middle| \begin{array}{c} \forall i \in [1,n],\ \vec{v}_i.\vec{x} = a_i \\ \forall j \in [1,m],\ \vec{w}_j.\vec{x} = b_j \end{array}\right\}\right| = q^{\alpha-(n+m)}$$

Then on the second set of solutions, we first remark that:

$$\begin{aligned} & \left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha, \vec{r} \in \mathbb{Z}_q^m, \middle| \begin{array}{c} \forall i \in [1,n],\ \vec{v}_i.\vec{x} = a_i \\ \forall j \in [1,m],\ \vec{r}_j = b_j \end{array}\right\}\right| \\ = & \left|\left\{\vec{x} \in \mathbb{Z}_q^\alpha, \middle| \forall i \in [1,n],\ \vec{v}_i.\vec{x} = a_i\right\}\right| \\ = & q^{\alpha-n} \end{aligned}$$

The conclusion is immediate from here. ■

An immediate corollary of this is that if $\vec{X}$ denotes a random variable over $\mathbb{Z}_q^\alpha$ (with uniform probability) and $\vec{R}$ denotes a random variable over $\mathbb{Z}_q^m$ (also with uniform probability). Then with the same hypothesis than in the previous proposition,

$$\begin{aligned} & Pr\left[\begin{array}{c} \forall i \in [n],\ \vec{v}_i.\vec{X} = a_i \\ \forall j \in [m],\ \vec{w}_j.\vec{X} = b_j \end{array}\right] \\ = & Pr\left[\begin{array}{c} \forall i \in [n],\ \vec{v}_i.\vec{X} = a_i \\ \forall j \in [m],\ \vec{R}_j = b_j \end{array}\right] \\ = & \frac{1}{q^{n+m}} \end{aligned}$$

Using this result, we now prove our result for a base case where all the monomials are variables.

Lemma 5 *Let x_1 to x_α be α different variables. Let $\mathcal{A}$ be an adversary against $G(x_1, \ldots, x_\alpha)$ then*

$$\mathrm{Adv}_{\mathcal{A}}^{G(x_1,\ldots,x_\alpha)} = 0$$

Proof: Let $\mathcal{A}$ be an adversary. Let us consider that $\mathcal{A}$ calls the first (resp. second) oracle n (resp. m) times with arguments $\vec{v}_i$ (resp. $\vec{w}_j$) and receives as output g^{a_i} (resp. g^{b_j}). Then as the families of vectors $\vec{v}$ and $\vec{w}$ are linearly independent, the probabilities of producing such observations with such arguments are the same in the two possible games. Therefore, we can compute the advantage of $\mathcal{A}$:

$$\begin{aligned} & \mathrm{Adv}_{\mathcal{A}}^{G(m_1,\ldots,m_\alpha)} \\ = & \mathbb{P}\left[G_{\mathcal{A}}^1(m_1,\ldots,m_\alpha) = 1\right] - \mathbb{P}\left[G_{\mathcal{A}}^0(m_1,\ldots,m_\alpha) = 1\right] \\ = & \sum_u \mathbb{P}\left[G_{\mathcal{A}}^1(m_1,\ldots,m_\alpha) = 1 | \mathcal{A} \text{ observed } u\right]\mathbb{P}\left[u\right] \\ & - \sum_u \mathbb{P}\left[G_{\mathcal{A}}^0(m_1,\ldots,m_\alpha) = 1 | \mathcal{A} \text{ observed } u\right]\mathbb{P}\left[u\right] \\ = & \sum_u \mathbb{P}\left[u\right].\big(\mathbb{P}\left[G_{\mathcal{A}}^1(m_1,\ldots,m_\alpha) = 1 | \mathcal{A} \text{ observed } u\right] \\ & - \mathbb{P}\left[G_{\mathcal{A}}^0(m_1,\ldots,m_\alpha\right] = 1 | \mathcal{A} \text{ observed } u]\big) \\ = & 0 \end{aligned}$$

■

The previous lemma is useful to initialize our induction argument. The next lemma allows to perform the induction itself.

Lemma 6 *Let m_1 to m_α be α monomials which set of variables is X_m. Let x_1 and x_2 be two different variables from X_m and let $x_{1,2}$ be a variable that is not in X_m. Let m'_1 to m'_α denote monomials such that m'_i is monomial m_i where $x_1.x_2$ is replaced by $x_{1,2}$.*

For any adversary $\mathcal{A}$ against $G(m_1, \ldots, m_\alpha)$, there exists an adversary $\mathcal{B}$ against DDH *such that:*

$$\mathrm{Adv}_{\mathcal{A}}^{G(m_1,\ldots,m_\alpha)} = 2.\mathrm{Adv}_{\mathcal{B}}^{\mathsf{DDH}} + \mathrm{Adv}_{\mathcal{A}}^{G(m'_1,\ldots,m'_\alpha)}$$

Proof: Adversary $\mathcal{B}$ is designed to play against DDH by using $\mathcal{A}$. It uses its DDH challenges to answer requests of $\mathcal{A}$. First it associates a group element g_i to each monomial m_i, this is done by function F that outputs a vector of values:

$$\begin{array}{lll} m \rightarrow exp(Z, m \setminus \{x_1, x_2\}) & & \text{if } x_1, x_2 \in m \\ m \rightarrow exp(X, m \setminus \{x_1\}) & & \text{else if } x_1 \in m \\ m \rightarrow exp(Y, m \setminus \{x_2\}) & & \text{else if } x_2 \in m \\ m \rightarrow exp(g, m) & & \text{else} \end{array}$$

Adversary $\mathcal{B}$ is given by:

Adversary $\mathcal{B}(X, Y, Z)$:
$\forall i, x_i \leftarrow \mathbb{Z}_q$
$b \leftarrow [0, 1]$
$g_1, \ldots, g_\alpha \leftarrow F(X, Y, Z, x_1, .., x_n)$
$d \leftarrow \mathcal{A}/\lambda\vec{v}. \prod_{i=1}^{\alpha} g_i^{\vec{v}[i]}$
$\quad\quad \lambda\vec{v}$.**if** b **then** $\prod_{i=1}^{\alpha} g_i^{\vec{v}[i]}$ **else** g^r
return $b = d$

Then the execution of $\mathcal{B}$ with arguments $(g^{x_1}, g^{x_2}, g^{x_1.x_2})$ is the same as the game that involves $\mathcal{A}$ against $G(m_1, \ldots, m_\alpha)$ whereas the execution of $\mathcal{B}$ with arguments $(g^{x_1}, g^{x_2}, g^{x_{1,2}})$ is the same as the game that involves $\mathcal{A}$ against $G(m'_1, \ldots, m'_\alpha)$. The advantage of $\mathcal{B}$ is given by:

$$\begin{aligned} 2\mathrm{Adv}_{\mathcal{B}}^{\mathsf{DDH}} &= 2Pr\big[\mathcal{B}(g^{x_1}, g^{x_2}, g^{x_1.x_2}) = 1\big] \\ &\quad -2Pr\big[\mathcal{B}(g^{x_1}, g^{x_2}, g^{x_{1,2}}) = 1\big] \\ &= \mathrm{Adv}_{\mathcal{A}}^{G(m_1,\ldots,m_\alpha)} - \mathrm{Adv}_{\mathcal{A}}^{G(m'_1,\ldots,m'_\alpha)} \end{aligned}$$

■

Finally, Proposition 12 can be proven by iterating this last lemma. For that purpose, we introduce $ord(\Gamma)$ which is defined as the sum of the order of monomials in Γ and proceed using an induction on $ord(\Gamma) - |\Gamma|$.

1. If $ord(\Gamma) - |\Gamma|$ equals 0, then all the monomials have order 1 and thus the number of variables α is equal to Γ. In that situation, Lemma 5 applies and gives us that for any adversary $\mathcal{A}$,
$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma)} = 0$$
2. Else $ord(\Gamma)$ is strictly greater than $|\Gamma|$ thus there exists a monomial m in Γ which order is strictly greater than 1. Let x_1 and x_2 be two variables from m. Let $x_{1,2}$ be a fresh variable and let m'_1 to m'_α denote monomials m_i where $x_1.x_2$ is replaced by $x_{1,2}$. The list of monomials m'_1 to m'_α is denoted by Γ'. Lemma 6 applies and hence,

   If $\mathcal{A}$ is an adversary against $G(m_1, \ldots, m_\alpha)$, there exists an adversary $\mathcal{B}_{|\Gamma|-\alpha}$ against DDH such that:
$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma)} = 2\mathrm{Adv}_{\mathcal{B}_{|\Gamma|-\alpha}}^{\mathsf{DDH}} + \mathrm{Adv}_{\mathcal{A}}^{G(\Gamma')}$$
   The induction hypothesis applies on Γ' (as the sum of orders have strictly decreased). Hence there exists $|\Gamma'| - (\alpha+1)$ adversaries against DDH denoted by $\mathcal{B}_1$ to $\mathcal{B}_{|\Gamma'|-(\alpha+1)}$ such that:
$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma')} = 2 \sum_{i=1}^{|\Gamma'|-(\alpha+1)} \mathrm{Adv}_{\mathcal{B}_i}^{\mathsf{DDH}}$$
   Moreover Γ' and Γ have the same number of elements hence we finally get that:
$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma)} = 2 \sum_{i=1}^{|\Gamma|-\alpha} \mathrm{Adv}_{\mathcal{B}_i}^{\mathsf{DDH}}$$
   We consider an adversary $\mathcal{B}(X, Y, Z)$ which randomly samples an integer i between 1 and $|\Gamma| - \alpha$ and executes $\mathcal{B}_i(X, Y, Z)$. Thus we obtain that:
$$\mathrm{Adv}_{\mathcal{A}}^{G(\Gamma)} = 2\left(|\Gamma| - \alpha\right)\mathrm{Adv}_{\mathcal{B}}^{\mathsf{DDH}}$$

■

Let us consider the case of α different variables. Then let us suppose that Γ contains all the monomials using the α variables. The number of such monomials is $2^\alpha - 1$. Hence, if the number of variables is polynomial in the security parameter η, then Γ contains an exponential number of elements. Even when DDH hold, Proposition 12 cannot be used to conclude that 3DH holds. As we do not want to consider an exponential number of monomials but we want to have a polynomial number of distinct variables, a solution is to bound the degree of the monomials independently of η.

Corollary 1 *Let P be a polynomial. Let x_1 to $x_{P(\eta)}$ be $P(\eta)$ different variables and m be an integer. Let Γ denote the list of monomials using x_i which orders are lower than m. If the* DDH *assumption holds, then the advantage of any adversary $\mathcal{A}$ against $G(\Gamma)$ is negligible.*

B.4.2 Soundness proof

The proof is done in three steps:

1. DDH $\Rightarrow \approx_{E_{\mathsf{DH}}}$ -ad-sound
2. $\approx_{E_{\mathsf{DH}}}$ -ad-sound $\Rightarrow \approx_{E_{\mathsf{DH}}}$ -sound; this is a direct application of Proposition 1.
3. $\approx_{E_{\mathsf{DH}}}$ -sound $\Rightarrow$ DDH; this has been proven in [10] for an extended theory.

We still have to prove step 1. Let $\mathcal{A}$ be an adversary against $\approx_{E_{\mathsf{DH}}}$-ad-soundness. We build two adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ against 3DH which uses $\mathcal{A}$ as a subroutine. As there is a bound l on the maximal order of monomials, the set of possible monomials Γ used by $\mathcal{A}$ has a polynomially bounded size and the reduction between 3DH and DDH is polynomial.

Let us fix bit b. Adversary $\mathcal{B}_b$ uses his RR oracle to answer queries made by $\mathcal{A}$.

Adversary $\mathcal{B}/\mathcal{O}_R, \mathcal{O}_{RR}$:
$d \leftarrow \mathcal{A}^{\mathcal{O}}$
return d

The implementation $\mathcal{O}$ of $\mathcal{A}$'s oracle stores queries made by $\mathcal{A}$ as well as the answers that it gives to $\mathcal{A}$. Let us suppose that $\mathcal{A}$ calls his oracle with pair (g^{p_i}, g^{q_i}) and has performed queries (g^{p_1}, g^{q_1}) to $(g^{p_{i-1}}, g^{q_{i-1}})$ before resulting in answers bs_1 to bs_{i-1}.

- If p_i is in $\mathrm{span}(p_1, \ldots, p_{i-1})$, then there exists a_1 to a_{i-1} such that:
$$p_i = \sum_{j=i}^{i-1} a_j p_j$$

As the adversary is legal, we also have that:

$$q_i = \sum_{j=i}^{i-1} a_j q_j$$

The answer given to $\mathcal{A}$ is $\prod_{j=1}^{i-1} bs_j^{a_j}$ where the multiplications and exponentiations concern elements of the group.

- Else p_i is not in $\mathrm{span}(p_1, \ldots, p_{i-1})$ and q_i is not in $\mathrm{span}(q_1, \ldots, q_{i-1})$. If $b = 0$, adversary $\mathcal{B}_0$ uses his RR oracle on p_i. Else if $b = 1$, adversary $\mathcal{B}_1$ uses his RR oracle on q_i. The result is given to $\mathcal{A}$.

When his challenge bit is 1, adversary $\mathcal{B}_b$ corresponds to $\mathcal{A}$ using oracle $\mathcal{O}^b_{LR,A_\eta}$. When their challenge bits are 0, adversaries $\mathcal{B}_0$ and $\mathcal{B}_1$ have the same behavior (as their RR oracle always answer with a random group element). Hence:

$$\begin{aligned} \mathbb{P}\left[\mathcal{A}^{\mathcal{O}^b_{LR,A_\eta}} = 1\right] &= \mathbb{P}\left[G^1_{\mathcal{B}_b}(\Gamma) = 1\right] \\ \mathbb{P}\left[G^0_{\mathcal{B}_0}(\Gamma) = 1\right] &= \mathbb{P}\left[G^0_{\mathcal{B}_1}(\Gamma) = 1\right] \end{aligned}$$

Thus we obtain that:

$$\mathrm{Adv}^{\Gamma}_{\mathcal{B}_1} - \mathrm{Adv}^{\Gamma}_{\mathcal{B}_0} = \mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A},A_\eta}$$

As DDH holds, the advantages of $\mathcal{B}_0$ and $\mathcal{B}_1$ are negligible so the advantage of $\mathcal{A}$ is also negligible and (A_η) is $\approx_{E_{\mathsf{DH}}}$-ad-sound.

B.5 Proof of Proposition 9

Proposition 9 *Let $\prec$ be a total order between keys and exponentiations. Let (A_η) be an EE-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

First, we justify a restriction on the queries made by adversaries. As the frames are well-formed we have that any subterm $\exp(p)$ used at a key position is linearly independent from any p' such that $\exp(p')$ is another subterm of the frame. Therefore, the advantage of any adversary $\mathcal{A}$ against adaptive soundness can be split into the advantage of two adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ against the dynamic variant of DDH, 3DH, introduced in Appendix B.4, plus the advantage of another adversary $\mathcal{B}_3$ against adaptive soundness such that, in queries of $\mathcal{B}_3$, only names of sort Data occur at key positions.

Lemma 7 *Let Γ be the (polynomial) set of all possible monomials. Let $\mathcal{A}$ be a legal adversary against $\approx_E$-ad-soundness. There exist two adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ against* 3DH *and a legal adversary $\mathcal{B}_3$ against $\approx_E$-ad-soundness such that:*

- *in queries of $\mathcal{B}_3$, only names of sort* Data *occur at key position;*
- *the advantages satisfies the following relation:*

$$\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A},A_\eta}(\eta) = \mathrm{Adv}^{\Gamma}_{\mathcal{B}_1}(\eta) + \mathrm{Adv}^{\Gamma}_{\mathcal{B}_2}(\eta) + \mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{B}_3,A_\eta}(\eta)$$

Proof: Subterms $\exp(p)$ used at a key position are linearly independent from any p' such that $\exp(p')$ is another subterm of the frame. Adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ use $\mathcal{A}$ as a sub-routine but answer to queries made by $\mathcal{A}$ in different ways.

- When $\mathcal{A}$ queries his oracle with (t_0, t_1), $\mathcal{B}_1$ only uses t_1. He generates all the necessary names of sort Data. For subterm $\exp(p)$ of t_1, the real oracle of $\mathcal{B}_1$ is used if $\exp(p)$ does not occur at a key position and has not occurred at a key position previously (moreover, as $\mathcal{A}$ is legal we know that $\exp(p)$ will not occur at a key position in further requests). In the other case, the RR oracle of $\mathcal{B}_1$ is used to obtain a bit-string for $\exp(p)$. Finally $\mathcal{B}_1$ performs the necessary operations using algorithms from (A_η) and returns the result to $\mathcal{A}$.
- $\mathcal{B}_2$ is similar to $\mathcal{B}_1$ but uses t_0 instead of t_1.
- When $\mathcal{A}$ queries his oracle with (t_0, t_1), $\mathcal{B}_3$ queries his own oracle with (t'_0, t'_1) where t'_0 and t'_1 are as t_0 and t_1 but subterms $\exp(p)$ used at key positions are replaced by fresh names of sort Data in a consistent way (*i.e.*, the same name is used for two occurrences $\exp(p)$ and $\exp(q)$ if $p =_{E_2} q$). The output of the oracle is returned to $\mathcal{A}$.

When the challenge bit of $\mathcal{B}_1$ is 1, the real bit-strings are used for $\exp(p)$ at key positions in t_1. This corresponds to the execution of $\mathcal{A}$ with challenge bit 1. When the challenge bit of $\mathcal{B}_1$ is 0, random bit-strings are used for $\exp(p)$ at key position. Hence this corresponds to the execution of $\mathcal{B}_3$ with challenge bit 1. In a similar way, when the challenge bit of $\mathcal{B}_2$ is 1, this corresponds to the execution of $\mathcal{B}_3$ with challenge bit 0. When the challenge bit of $\mathcal{B}_2$ is 0, this corresponds to the execution of $\mathcal{A}$ with challenge bit 0. Thus we obtain the expected relation among advantages:

$$\mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{A},A_\eta}(\eta) = \mathrm{Adv}^{\Gamma}_{\mathcal{B}_1}(\eta) + \mathrm{Adv}^{\Gamma}_{\mathcal{B}_2}(\eta) + \mathrm{Adv}^{\mathsf{ADPT}}_{\mathcal{B}_3,A_\eta}(\eta)$$

■

As DDH holds, the advantages of adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ against 3DH are negligible because of proposition 12. Without loss of generality, we restrict ourselves to frames where only names (of sort Data) occur at key positions. Any occurrence of $\exp(p)$ at a key position can be replaced by a fresh name.

We consider the function σ which is defined recursively on the last element t^i of its input list $[t^1...t^i]$ by the following algorithm p:

$$\begin{array}{rcll} p((t',t'')) & = & (p(t'),p(t'')) & \\ p(\{t'\}_k) & = & \{p(t')\}_k & \text{if } \{x_j \mapsto t_j\}_{1\leq j\leq i} \vdash_{E_1\cup E_2} k \\ p(\{t'\}_k) & = & \{k'\}_k & \text{if } \{x_j \mapsto t_j\}_{1\leq j\leq i} \nvdash_{E_1\cup E_2} k \\ p(t') & = & t' & \text{otherwise} \end{array}$$

where k' is a fresh name of sort Data. We define $\sigma([t^1...t^i])$ to be $p(t^i)$. Note that $p(t)$ is similar to the pattern of term t in the sense of [4].

Let F be the maximal set of pairs of frames for which σ is a (E_1, E_2)-hybrid function. We first prove soundness for pairs of frames in F.

Lemma 8 *Let (A_η) be an EE-secure family of computational algebras. (A_η) is $\approx_E$-ad-sound for pairs of frames in F.*

Proof: As (A_η) is an EE-secure family of computational algebras, the symmetric encryption in (A^1_η) is IND-CPA, and the group in (A^2_η) satisfies DDH. Moreover, if φ is well-formed for $\prec$, then $\sigma(\varphi)$ is also well-formed for $\prec$. Using the proof of Proposition 5 (given in Appendix B.2), we obtain that $A^1_\eta \times A^2_\eta$ is $\approx_{E_1}$-ad-sound for pairs of frames $(\sigma(\varphi), \varphi)$. By Proposition 8, A^2_η is $\approx_{E_2}$-ad-sound for frames on Σ_2. Therefore this lemma is as a consequence of Proposition 3. ∎

In order to obtain soundness for all pairs of well-formed frames, rather than only those in F, we introduce a function ρ which takes as input two lists of terms $[t^1_1, ..., t^i_1]$ and $[t^1_0, ..., t^i_0]$, such that $\{x_i \mapsto t^j_0\}_{1\leq j\leq i} \approx_{E_1\cup E_2} \{x_i \mapsto t^j_0\}_{1\leq j\leq i}$, and outputs a substitution of keys $\mu(t^i_1, t^i_0)$ where μ is inductively defined on its parameters (t, u):

- If t is a pair (t', t'') and u is a pair (u', u''), the output of μ is $\mu(t', u') \cup \mu(t'', u'')$.
- If t is an encryption $\{t'\}_k$ and u is an encryption $\{u'\}_{k'}$ where k and k' are not deducible, then μ maps k to k'.
- If t is an encryption $\{t'\}_k$ and u is an encryption $\{u'\}_{k'}$ where k and k' are deducible, then μ is recursively applied to (t', u') and maps k to k'.
- If t and u are names k and k' of sort Data, μ maps k to k'.
- Else μ outputs the empty substitution.

An important point is that ρ can be computed in polynomial time. We extend ρ to be a function from pairs of statically equivalent frames to frames in the following way:

$$\rho\left(\{x_i \mapsto t^i_1\}_i, \{x_i \mapsto t^i_0\}_i\right) = \{x_i \mapsto t^i_1\rho([t^1_1, ..., t^i_1], [t^1_0, ..., t^i_0])\}_i$$

Lemma 9 *Let φ and φ' be two well-formed frames for $\prec$ such that $\varphi_1 \approx_{E_1\cup E_2} \varphi_2$. Then distributions $[\![\varphi]\!]_{A_\eta}$ and $[\![\rho(\varphi, \varphi')]\!]_{A_\eta}$ are equal. Moreover the pair $(\rho(\varphi, \varphi'), \varphi')$ is in F.*

Proof: As $\varphi = \{x_i \mapsto t^i_1\}_i$ and $\varphi' = \{x_i \mapsto t^i_0\}_i$ are statically equivalent we have that $\rho([t^1_1, ..., t^i_1], [t^1_0, ..., t^i_0])$ is a bijective renaming of names. Hence, distributions $[\![\varphi]\!]_{A_\eta}$ and $[\![\rho(\varphi, \varphi')]\!]_{A_\eta}$ are trivially equal.

We now have to prove that the pair $(\rho(\varphi, \varphi'), \varphi')$ is in F. For this we need to show that σ is a hybrid function for this pair of frames. First, note that for any frame φ, $\varphi \approx_{E_1} \sigma(\varphi)$. Hence the first point defining hybrid functions is satisfied by σ. The second point is satisfied by definition of ρ and σ: ρ makes the necessary renamings such that Σ_1 positions (and the root of the terms at these positions) of $\sigma(\rho(\varphi, \varphi'))$ and $\sigma(\varphi')$ coincide. Moreover, all Σ_2 minimal positions in $\sigma(\rho(\varphi, \varphi'))$ and in $\sigma(\varphi')$ are accessible, *i.e.*, for any such position p and its associated subterm t, there exists a term M such that $M\varphi =_{E_1\cup E_2} t$ (applying σ replaces any such inaccessible position by a fresh name). Hence the frame containing all the subterms at these positions for $\sigma(\rho(\varphi, \varphi'))$ and the frame containing all the subterms at these positions for $\sigma(\varphi')$ are statically equivalent for E_2. ∎

Using Lemmas 8 and 9, it is now possible to prove our adaptive soundness result. Let $\mathcal{A}$ be an adversary against adaptive soundness for well-formed frames for $\prec$. We introduce a new implementation of oracle $\mathcal{O}^b_{LR,A_\eta}$: $\mathcal{H}_{A_\eta}$. When called on the sequence of queries $(t^j_0, t^j_1)_{1\leq j\leq i}$, $\mathcal{H}_{A_\eta}$ outputs $[\![t^i_1\rho([t^1_1, ..., t^i_1], [t^1_0, ..., t^i_0])]\!]_{A_\eta}$.

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

As a direct consequence of Lemma 9 (distributions are identical) the first difference of probability equals zero. In the second difference of probabilities, the left-right interpretation is performed on a pair of frames in F because of Lemma 9. Moreover, due to Lemma 8, in this case the advantage of an adversary is negligible. Hence the advantage of $\mathcal{A}$ is negligible which allows us to conclude that (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.

B.6 Proof of Proposition 10

Proposition 10 *Let $\prec$ be a total order between keys and terms of sort $\mathrm{Data}_{\oplus}$. Let (A_η) be an EX-secure family of computational algebras then (A_η) is $\approx_E$-ad-sound for well-formed frames for $\prec$.*

This proof closely follows the proof of Proposition 9, given in Appendix B.5, and can easily be adapted.

C Proof sketch of Proposition 11

Proposition 11 *Let (A_η) be a family of computational algebras and $\Pi = (\mathcal{S}, \mathcal{J}, \mathcal{L}, \mathcal{K})$ be a DKE. If (A_η) is $\approx_E$-ad-sound and Π is secure in the symbolic model, then Π is secure in the concrete model.*

Let $\Pi = (\mathcal{S}, \mathcal{J}, \mathcal{L}, \mathcal{K})$ be a DKE and $\mathcal{A}$ be an adversary against its security. We build an adversary $\mathcal{B}$ against adaptive soundness for (A_η) that uses $\mathcal{A}$ as a subroutine and which advantage is comparable to the advantage of $\mathcal{A}$. $\mathcal{B}$ locally stores the state of the protocol through a triple $\langle L, C, T\rangle$ and answers the queries made by $\mathcal{A}$ to his oracles in the following way:

- For any query $\mathsf{Corrupt}(U)$ made by $\mathcal{A}$, $\mathcal{B}$ adds U to his set of corrupted users C.
- For any query $\mathsf{Setup}(U_1, \ldots, U_n)$ made by $\mathcal{A}$, $\mathcal{B}$ uses the $\mathcal{S}$ algorithm on $(U_1, \ldots, U_n)$ and obtains a list of formal terms t_1 to t_m. U_1 to U_n are added to the set of users U. t_1 to t_m are added to the set of sent formal terms T. Let t'_i be as t_i but nonces from corrupted users have been removed, $t'_i = t_i c_1 \ldots c_j$. $\mathcal{B}$ uses his left-right evaluation oracle on pairs (t'_i, t'_i) for each i and obtains bit-strings bs_i. Then $\mathcal{B}$ randomly samples values (in $\mathbb{Z}_q$) for nonces from corrupted users c_k in the different t_i and exponentiates bs_i using the value of $c_1 \ldots c_j$. Finally, the evaluations of the different t_i are returned to $\mathcal{A}$.
- For any query $\mathsf{Join}(U_1, \ldots, U_n)$ made by $\mathcal{A}$, $\mathcal{B}$ uses the $\mathcal{J}$ algorithm on $(U_1, \ldots, U_n)$ and obtains a list of formal terms t_1 to t_m. U_1 to U_n are added to the set of users U. t_1 to t_m are added to the set of sent formal terms T. The same process as for the Setup oracle is applied to the t_i in order to get their evaluations and return them to $\mathcal{A}$.
- For any query $\mathsf{Leave}(U_1, \ldots, U_n)$ made by $\mathcal{A}$, $\mathcal{B}$ uses the $\mathcal{L}$ algorithm on $(U_1, \ldots, U_n)$ and obtains a list of formal terms t_1 to t_m. U_1 to U_n are removed from the set of users U. t_1 to t_m are added to the set of sent formal terms T.The same process as for the Setup oracle is applied to the t_i in order to get their evaluations and return them to $\mathcal{A}$.
- For any query Test made by $\mathcal{A}$, $\mathcal{B}$ uses the $\mathcal{K}$ algorithm to get the term t representing the group key. No nonce of corrupted user can occur in t so $\mathcal{B}$ submits the pair $(\exp(r), t)$ to his left-right evaluation oracle where r is a fresh nonce. The output of the oracle is given back to $\mathcal{A}$.

We supposed that protocol Π is formally secure so $\mathcal{B}$ is a legal adversary. Moreover when the challenge bit of $\mathcal{B}$ is b, $\mathcal{B}$ perfectly simulates the execution of $\mathcal{A}$ with oracle $\mathcal{O}_b$ hence the advantages of $\mathcal{A}$ against protocol Π and of $\mathcal{B}$ against adaptive soundness are equal. As (A_η) is $\approx_E$-ad-sound, the advantage of $\mathcal{B}$ is negligible so the advantage of $\mathcal{A}$ is also negligible and protocol Π is secure in the concrete model.