Soundness and Completeness of Formal Encryption: the Cases of Key Cycles and Partial Information Leakage

Pedro Adão[1]*, Gergei Bana[2]**, Jonathan Herzog[3], and Andre Scedrov[4]***

[1] SQIG–Instituto de Telecomunicações and IST, TULisbon, Portugal
[2] Department of Computer Science, UCDavis, USA
[3] The Naval Postgraduate School, Monterey, CA, USA
[4] Department of Mathematics, University of Pennsylvania, Philadelphia, USA

`pedro.adao@ist.utl.pt` `gebana@cs.ucdavis.edu` `jcherzog@nps.edu` `scedrov@math.upenn.edu`

Abstract. In their seminal work, Abadi and Rogaway [2, 3] show that the formal (Dolev-Yao) notion of indistinguishability is *sound* with respect to the computational model: messages that are indistinguishable in the formal model become indistinguishable messages in the computational model. However, this result leaves two problems unsolved. First, it cannot tolerate key cycles. Second, it makes the too-strong assumption that the underlying cryptography hides all aspects of the plaintext, including its length. In this paper we extend their work in order to address these problems.

We show that the recently-introduced notion of KDM-security can provide soundness even in the presence of key cycles. For this, we have to consider encryption that reveals the length of plaintexts, which we use to motivate a general examination information-leaking encryption. In particular, we consider the conditions under which an encryption scheme that may leak some partial information will provide soundness and completeness to some (possibly weakened) version of the formal model.

1 Introduction

Historically, cryptographic protocols have been studied and analyzed in at least two different models. The first of these models, the *computational model*, is derived from

* Partially supported by FCT grant SFRH/BD/8148/2002. Additional support from FEDER/FCT project QuantLog POCI/MAT/55796/2004, FEDER/FCT project QSec PTDC/EIA/67661/2006, and FEDER/FCT project KLog PTDC/MAT/68723/2006.

** Partially supported by OSD/ONR CIP/SW URI 'Software Quality and Infrastructure Protection for Diffuse Computing" through ONR Grant N00014-01-1-0795. Additional support from NSF Grant CNS-0429689. Additional support from the Packard Fellowship. Part of this work was done while the author was affiliated with University of Pennsylvania, Department of Mathematics.

*** Partially supported by OSD/ONR CIP/SW URI "Software Quality and Infrastructure Protection for Diffuse Computing" through ONR Grant N00014-01-1-0795 and OSD/ONR CIP/SW URI "Trustworthy Infrastructure, Mechanisms, and Experimentation for Diffuse Computing" through ONR Grant N00014-04-1-0725. Additional support from NSF Grants CCR-0098096 and CNS-0429689.

complexity theory. Its definitions are phrased in terms of the asymptotic behavior of Turing machines, and its main proof technique is the reduction. The other of these two models, the *formal model (or, Dolev-Yao model)*, is so named because of its genesis in the field of formal methods. Its definitions are phrased in terms of process algebras and state machines (particularly non-deterministic ones) and it uses many different proof methods (including automated ones).

In this work (based on [5, 4, 14]) we consider two aspects of these models' relationship. The differences between these models are many, but two in particular are key: their representations of messages and the powers they give to the adversary.

– In the computational model, messages are families of probability distributions over bit-strings (indexed by the security parameter). The adversary is modeled as an algorithm of realistic computational power: probabilistic polynomial-time, PPT.
– The formal model imposes a great deal more structure. Messages are expressions built according to a particular grammar. Atomic messages are symbols representing keys, random values, texts, and so on. More complex messages can be built from simpler ones by application of (symbolic) functions, e.g., pairing and encryption. The adversary is only given limited power to manipulate these expressions, such as separating a concatenation or decrypting an encryption (if it knows the decrypting key). These possible operations are specified via a set of equations.

Despite these differences, certain intuitions can be translated between the two models in the expected way. In particular, under carefully chosen conditions, *indistinguishability of messages* can be mapped directly from one model to the other. This was first demonstrated by Abadi and Rogaway [2, 3] in a particular setting and under strong assumptions. In their formulation of the formal model, two expressions are thought to be indistinguishable to the adversary, also called *formally equivalent*, if their only differences lie in encryption terms that cannot be decrypted by the formal adversary. In the computational model, on the other hand, messages are families of probability distributions on bit-strings. Indistinguishability of computational messages is captured by the standard notion of computational indistinguishability (*i.e.*, indistinguishability by an efficient algorithm).

Relating the two models. Once a computational encryption scheme is fixed, an intuitive function translates expressions between the two models. This function (called *interpretation*), maps each formal expression to an ensemble (indexed by the security parameter) of probability distributions over bit-strings. Given an encryption scheme, and hence a particular interpretation function, one can then ask whether all pairs of equivalent formal messages map to indistinguishable probability distribution ensembles. If so, it is said that *soundness* holds[5] and it implies that the formal model is a faithful abstraction of the computational model: security in the formal model implies security in the computational model as well.

In their seminal work, Abadi and Rogaway demonstrated that (in the symmetric-key encryption setting) soundness holds when the security level of the computational

[5] This particular kind of soundness is but one piece of a much larger definition, but as a convenient shorthand we will use 'soundness' in this paper to mean soundness of message indistinguishability.

encryption scheme is 'type-0,' a property of their own devising. This result was later translated to the public-key setting by Micciancio and Warinschi [46], who found that soundness is guaranteed by encryption schemes that satisfy 'chosen-ciphertext security' [51, 52] (CCA-2 in the notation of [18]). This power of chosen-ciphertext security has been confirmed by subsequent extensions [33, 23]. These results, however—in both the symmetric and asymmetric settings—do not address two important problems.

Unsolved problems in previous soundness results. Firstly, none of the existing soundness results address the problem of key cycles. An expression has a (symmetric) key cycle if one can find symmetric keys K_1, K_2,..., K_n such that K_i is encrypted in the expression under K_{i+1} and K_n is encrypted by K_1. (In the asymmetric setting, the public key K_{i+1} encrypts the private key K_i^{-1}, and K_1 encrypts K_n^{-1}.) The formal model makes no distinction between those messages that have key cycles and those that do not. Further, the interpretation function is well-defined over key cycles, and so, formal key cycles are computationally meaningful. However, neither the soundness result of Abadi and Rogaway nor subsequent soundness results (described in Section 2) are known to hold for such messages. (In fact, the stronger of these results [11, 23] assumes that no private or symmetric keys are encrypted at all!)

Another problem that was not dealt in most of the previous soundness results regards to partial leakage of information. Most of these results consider that formal encryption hides all information about the plaintext. As an example, the original Abadi and Rogaway result assumes that formal encryption conceals all aspects of the plaintext. That is, their result requires that symmetric encryption hides (among other things) the length of the plaintext. Unfortunately, this cannot be achieved except in very limited contexts. This particular issue has been noted by Backes, Pfitzmann and Waidner [13], and Backes and Pfitzmann [8]. Furthermore, it is the focus of work by Micciancio and Warinschi [46], Laud [39], and Micciancio and Panjwani [44] who resolve the matter by weakening the formal model. These results, however, are highly specific to particular classes of computational encryption schemes. Can these results can be generalized to encompass other encryption schemes that leak other kinds of information? Rephrased, under what conditions will an encryption scheme provide soundness to some formal model?

1.1 Our work

In this paper, we extend the original result of Abadi and Rogaway in order to address the problems mentioned above. First, we extend the formalism of Abadi Rogaway and show that soundness in the presence of key cycles can actually be achieved using a recently-proposed notion of computational security. In doing this, however, we must (unlike Abadi and Rogaway) assume that formal encryptions reveal two things: the 'length' of their plaintexts, and whether two different ciphertexts were created using the same key. With this as motivation, we then turn to generalizations of the Abadi-Rogaway formalism. In particular, we show (in a general way) how Abadi and Rogaway's formulation of the formal model can be extended to consider encryption schemes (computational or information-theoretic) that leak partial information such as plaintext-length. That is, we investigate those conditions under which a computational encryption scheme provides soundness and completeness to some (possibly weakened) version of the formal model.

In more detail: We resolve the issue of soundness in the presence of key cycles by using the notion of *key-dependent message* (KDM) security for symmetric encryption. This definition was recently introduced simultaneously both by Black, Rogaway and Shrimpton [19], who consider it in their own right, and by Camenisch and Lysyanskaya [20], who use it for an anonymous credential system. We, however, will use it to demonstrate two points:

1. As expected, and predicted by Black *et al.*, this new definition is strong enough to provide soundness in the presence of keys cycles.
2. Moreover, soundness *requires* new computational definitions of security. That is, we demonstrate that both soundness and KDM security neither imply nor are implied by type-0 security, the notion of security used by Abadi and Rogaway.

Thus, the problem of key cycles was, in fact, a genuine "gap" between the formal and computational models at the time of the original Abadi-Rogaway result, but one that can be repaired using recent advances in the computational model. Also, soundness in the presence of key cycles demonstrates that there is more to the relationship between the formal and computational models than type-0.

Unfortunately, KDM-secure encryption does not necessarily hide all aspects of its inputs. In particular, KDM-security allows a ciphertext to reveal two things: the bit-length of the plaintext, and the identity (but not value) of the key used in the encryption. Therefore, soundness for key cycles requires that encryptions in the formal model must also reveal these two things.

This fact leads us to another extension of the original Abadi-Rogaway result. Their result assumes that computational encryption can hide *all* aspects of the plaintext. In particular, it demonstrates that soundness is provided by 'type-0' encryption, which hides (among other things) the length of the plaintext. However, most available encryption schemes do not hide this information. For this reason, the original Abadi-Rogaway result should be generalized to consider the kinds of soundness that can be provided by real encryption schemes.

The Problem of Leakage of Partial Information More specifically, we extend the applicability of the Abadi-Rogaway treatment by expanding their formulation of the formal model. We show how to adjust the formal notion of equivalence in order to maintain soundness when the underlying computational encryption scheme leaks partial information. Furthermore, we investigate the circumstances under which an encryption scheme (or security definition) can be thought of as implementing *a* (possibly weakened) version of the formal model.

Also, our approach captures both the standard complexity-based encryption schemes of the computational model and purely probabilistic, *information-theoretic* encryption schemes. That is, we use a general probabilistic framework that includes, as special cases, both the computational and purely probabilistic encryption schemes (such as One-Time Pad).

We consider not only soundness properties, but we also provide *completeness* theorems. In this context, an encryption scheme provides soundness if, when used in the interpretation function, equivalent formal messages become indistinguishable probability distributions. On the other hand, a scheme provides completeness if whenever two formal messages have indistinguishable interpretations, they are equivalent in the formal model. Our generalization will show how both of these conditions can be maintained. Since key cycles do not pose a problem for completeness, we will only discuss completeness regarding the leak of information.

2 Previous Work

Work intended to connect the cryptographic and the formal models started with several independent approaches, including Lincoln, Mitchell, Mitchell, and Scedrov [41], Canetti [22], Pfitzmann, Schunter and Waidner [48, 49], and Abadi and Rogaway [3]. In [3], formal terms with nested operations are considered specifically for symmetric encryption, the adversary is restricted to passive eavesdropping, and the security goals are formulated as indistinguishability of terms. This was extended in [1] from terms to more general programs, but the restriction to passive adversaries remained. We discuss other extensions of [3] further below. Several papers consider specific models or specific properties, *e.g.,* Guttman, Thayer, and Zuck [31] consider strand spaces and information-theoretically secure authentication.

A process calculus for analyzing security protocols in which protocol adversaries may be arbitrary probabilistic polynomial-time processes is introduced in [41]. In this framework, which provides a formal treatment of the computational model, security properties are formulated as observational equivalences. Mitchell, Ramanathan, Scedrov, and Teague [47] use this framework to develop a form of process bisimulation that justifies an equational proof system for protocol security.

The approach by Pfitzmann, Schunter and Waidner [48, 49] starts with a general reactive system model, a general definition of cryptographically secure implementation by simulatability, and a composition theorem for this notion of secure implementation. This work is based on definitions of secure *function* evaluation, *i.e.*the computation of one set of outputs from one set of inputs [29, 43, 16, 21]. The approach was extended from synchronous to asynchronous systems in [50, 22], which are now known as the *reactive simulatability framework* [50, 10] and the *universal composability framework* [22]. A detailed comparison of the two approaches may be found in [27].

The first soundness result of a formal model under active attacks has been achieved by Backes, Pfitzmann and Waidner [11] within the reactive simulatability framework. Their result comprises arbitrary active attacks and holds in the context of arbitrary surrounding interactive protocols and independently of the goals that one wants to prove about the surrounding protocols; in particular, property preservation theorems for the simulatability have been proved, *e.g.,* for integrity and secrecy [6, 9]. While the original result in [11] considered public-key encryption and digital signatures, the soundness result was extended to symmetric authentication and to symmetric encryption in [12] and [8], respectively. (These authors are also among the first to explicitly note that symbolic models of cryptography ignore plaintext-lengths while real cryptographic algorithms often reveal it [13, 8].)

Concurrently with [11], an extension to asymmetric encryption, but still under passive attacks, is in [34]. Asymmetric encryption under active attacks is considered in [32] in the random oracle model. Laud [39] has subsequently presented a cryptographic underpinning for a formal model of symmetric encryption under active attacks. His work enjoys a direct connection with a formal proof tool, but it is specific to certain confidentiality properties and restricts the surrounding protocols to straight-line programs in a specific language. Herzog *et al.* [34] and Micciancio and Warinschi [46] also give a cryptographic underpinning under active attacks. Their results are narrower than that in [11] since they are specific for public-key encryption, but consider simpler real implementations. Moreover, [34] relies on a stronger assumption, which was subsequently weakened by Herzog [33]. The approach in [46] restricts the classes of protocols and protocol properties that can be analyzed. The work of [46] was subsequently extended by Micciancio and Panjwani [44] to prove soundness of a group-key distribution protocol in the presence of a CPA-secure scheme. Cortier and Warinschi [24] use automated tools for proving that symbolic integrity and specific secrecy proofs are sound with respect to the computational model in the case of protocols that use nonces, signatures and asymmetric encryption (see below for the relationship between symbolic and cryptographic secrecy). Bana [14] and Adão, Bana, and Scedrov [5] extend the original Abadi-Rogaway result to weaker encryption schemes. Laud and Corin [40] consider extensions to composite keys, while Baudet, Cortier, and Kremer [15] consider extensions to equational theories and to static equivalence.

Impagliazzo and Kapron [36] suggest a formal logic for reasoning about probabilistic polynomial-time indistinguishability. Datta, Derek, Mitchell, Shmatikov, and Turuani [26] describe a cryptographically sound formal logic for proving protocol security properties without explicitly reasoning about probability, complexity, or the actions of a malicious attacker.

Recently, there has been concurrent and independent work on linking symbolic and cryptographic secrecy properties. Cortier and Warinschi [24] have shown that symbolically secret nonces are also computationally secret, *i.e.*, indistinguishable from a fresh random value given the view of a cryptographic adversary. Backes and Pfitzmann [9] and Canetti and Herzog [23] have established new symbolic criteria that suffice to show that a key is cryptographically secret. Backes and Pfitzmann formulate this as a property preservation theorem from the formal model to a concrete implementation while Canetti and Herzog link their criteria to ideal functionalities for mutual authentication and key exchange protocols. Backes and Pfitzmann have additionally provided a new definition of secrecy of payloads, *i.e.*application data, in a reactive framework, and they give sufficient symbolic criterion for this definition.

The first cryptographically sound security proofs of the Needham-Schroeder-Lowe protocol have been presented concurrently and independently in [7] and [53]. While the first paper conducts the proof within a deterministic, symbolic framework, the proof in the second paper is done from scratch in the cryptographic approach; on the other hand, the second paper proves stronger properties and further shows that chosen-plaintext-secure encryption is insufficient for the security of the protocol.

Regarding completeness, Micciancio and Warinschi [45] showed that a sufficiently strong encryption scheme enforces completeness for indistinguishability properties, and later Horvitz and Gligor [35] strengthened this result by giving an exact characterization of the computational requirements on the encryption scheme under which completeness holds. Later, it was shown by Bana [14] and Adão, Bana, and Scedrov [5] that completeness also holds for a more general class of (weaker) encryption systems. We only briefly mention that the simulatability-based results of [11, 12, 8] have shown completeness implicitly to establish the notion of simulatability.

We stress that none of the aforementioned soundness results hold in the presence of key cycles. The problem of soundness in the presence of key cycles was already addressed by Laud [38]. Laud's solution provides soundness in the presence of key cycles, but does so by weakening the notion of formal equivalence. It is assumed that key cycles somehow always 'break' the encryption and the formal adversary is strengthened so as to be always able to 'see' inside the encryptions of a key cycle. Soundness in the presence of key cycles naturally holds under this assumption, but we feel that the price paid is too high. Formal equivalence should reflect the ability of the formal adversary to distinguish messages, which should in turn reflect the actual extent to which the computational adversary can distinguish messages. It is often unreasonable from a cryptographer's point of view to *a priori* assume that the computational adversary can break all key cycles. We therefore propose, in this work, to demonstrate soundness in the presence of key cycles not by weakening encryption in the formal model, as suggested by Laud, but by strengthening it in the computational one.

Lastly, we hasten to point out that this work was the direct result of two previous conference papers [5, 4] and a PhD thesis [14] by the same authors. Although our previous treatment of key cycles [4] considered asymmetric encryption, it is overwhelmingly similar to the treatment of symmetric encryption to be found here.

3 The Abadi-Rogaway Soundness Theorem

In this section, we provide the context and the basic notions for our work. We do this by briefly summarizing the main definitions and results of Abadi and Rogaway's original work [2, 3]. In particular, we start presenting the formal model, then describe the computational model, and then introduce the notions of soundness and completeness.

3.1 The Formal Model

In this model, messages (or *expressions*) are defined at a very high level of abstraction. The simplest expressions are symbols for atomic keys and bit-strings. More complex expressions are created from simpler ones via encryption and concatenation, which are defined as abstract, 'black-box' constructors.

Definition 1 (Symmetric Expressions). *Let* ***Keys*** $= \{K_1, K_2, K_3, \dots\}$ *be an infinite discrete set of symbols, called the set of symmetric keys. Let* ***Blocks*** *be a finite subset of* $\{0,1\}^*$. *We define the* set of expressions, ***Exp***, *by the grammar:*

$$\textbf{\textit{Exp}} ::= \textbf{\textit{Keys}} \;\mid\; \textbf{\textit{Blocks}} \;\mid\; (\textbf{\textit{Exp}}, \textbf{\textit{Exp}}) \;\mid\; \{\textbf{\textit{Exp}}\}_{\textbf{\textit{Keys}}}$$

Let ***Enc*** $::= \{\textbf{\textit{Exp}}\}_{\textbf{\textit{Keys}}}$. *We will denote by* $\mathit{Keys}(M)$ *the set of all keys occurring in* M. *Expressions of the form* $\{M\}_K$ *are called* encryption terms.

Expressions may represent either a single message sent during an execution of the protocol, or the entire knowledge available to the adversary. In this second case, the expression contains not only the messages sent so far, but also any additional knowledge in the adversary's possession.

We wish to define when two formal expressions are indistinguishable to the adversary. Intuitively, this occurs when the only differences between the two messages lie within encryption terms that the adversary cannot decrypt. In order to rigorously define this notion, we first need to formalize when an encryption term is 'undecryptable' by the adversary, which in turn requires us to define the set of keys that the adversary can learn from an expression.

An expression might contain keys in the clear. The adversary will learn these keys, and then be able to use them to decrypt encryption terms of the expression—which might reveal yet more keys. By repeating this process, the adversary can learn the set of *recoverable decryption keys*:

Definition 2 (Subexpressions, Visible Subexpressions, Recoverable Keys, B-Keys, Undecryptable Terms). *We define the* set of subexpressions *of an expression M, denoted by $sub\,(M)$, as the smallest subset of expressions such that:*

- $M \in sub\,(M)$,
- $(M_1, M_2) \in sub\,(M) \implies M_1 \in sub\,(M)$ *and* $M_2 \in sub\,(M)$, *and*
- $\{M'\}_K \in sub\,(M) \implies M' \in sub\,(M)$.

We say that N is a subexpression of M, and denote it by $N \sqsubseteq M$, if $N \in sub\,(M)$.

The set of visible subexpressions *of a symmetric expression M, $vis\,(M)$, is the smallest subset of expressions such that:*

- $M \in vis\,(M)$,
- $(M_1, M_2) \in vis\,(M) \implies M_1 \in vis\,(M)$ *and* $M_2 \in vis\,(M)$, *and*
- $\{M'\}_K$ *and* $K \in vis\,(M) \implies M' \in vis\,(M)$.

The recoverable keys of *a (symmetric) expression M, R-Keys(M), are those that an adversary can recover by looking at an expression. That is, R-Keys$(M) = vis\,(M) \cap$ Keys(M).*

We say that an encryption term $\{M'\}_K \in vis\,(M)$ is undecryptable *in M if $K \notin$ R-Keys(M). Among the non-recoverable keys of an expression M, there is an important subset denoted by B-Keys(M). The set B-Keys(M) contains those keys which encrypt the outermost undecryptable terms. Formally, for an expression M, we define B-Keys(M) as*

$$\textit{B-Keys}(M) = \{K \in \textit{Keys}(M) \mid \{M\}_K \in vis\,(M) \ \text{ but } K \notin \textit{R-Keys}(M)\}\,.$$

Example 3. Let M be the following expression

$$((\{0\}_{K_6}, \{\{K_7\}_{K_1}\}_{K_4}), ((K_2, \{(\{001\}_{K_3}, \{K_6\}_{K_5})\}_{K_5}), \{K_5\}_{K_2})).$$

In this case, *Keys*$(M) = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7\}$. The set of recoverable keys of M is *R-Keys*$(M) = \{K_2, K_5, K_6\}$, because an adversary sees the non-encrypted K_2, and with that he can decrypt $\{K_5\}_{K_2}$, hence recovering K_5; then, decrypting twice with K_5, K_6 can be revealed. We also have that *B-Keys*$(M) = \{K_3, K_4\}$.

The formal model allows expressions to contain *key cycles*:

Definition 4 (Key Cycles). *We say that a set of keys* $\{L_1, \ldots, L_n\}$ *is* cyclic in an expression M*, if* M *contains encryption terms* $\{M_1\}_{L_1}$*,* $\{M_2\}_{L_2}$*, …,* $\{M_n\}_{L_n}$ *and* $L_{i+1} \sqsubseteq M_i$ *and* $L_1 \sqsubseteq M_n$*. In this case we say that we have a key cycle of length* n*. We will say that* M contains a key cycle *if there is a set of keys that is cyclic in* M*.*

According to our definition, expressions such as $\{\{M\}_K\}_K$ are not considered cyclic. As we will see, the original result of Abadi and Rogaway does not apply to expressions with key cycles. We extend their formalism in order to obtain soundness in the presence of key cycles.

3.2 Equivalence of Formal Expressions

A visible encryption term will appear 'opaque' to the adversary if and only if it is protected by at least one non-recoverable key. Thus, we wish to say that two expressions are equivalent if they differ only in the contents of their 'opaque' encryption terms. To express this, Abadi and Rogaway define the *pattern* of an expression through which equivalence of expressions will be obtained:

Definition 5 (Pattern (Classical)). *We define the* set of patterns*,* ***Pat****, by the grammar:*

$$\textbf{\textit{Pat}} ::= \textbf{\textit{Keys}} \mid \textbf{\textit{Blocks}} \mid (\textbf{\textit{Pat}}, \textbf{\textit{Pat}}) \mid \{\textbf{\textit{Pat}}\}_{\textbf{\textit{Keys}}} \mid \Box$$

The pattern of an expression M*, denoted by pattern*(M)*, is derived from* M *by replacing each encryption term* $\{M'\}_K \in vis\,(M)$ *(where* $K \notin$ *R-Keys*(M)*) by* $\Box$

For two patterns P *and* Q*,* $P = Q$ *is defined the following way:*

- *If* $P \in$ ***Blocks*** $\cup$ ***Keys****, then* $P = Q$ *iff* P *and* Q *are identical.*
- *If* P *is of the form* $\Box$*, then* $P = Q$ *iff* Q *is of the form* $\Box$
- *If* P *is of the form* (P_1, P_2)*, then* $P = Q$ *iff* Q *is of the form* (Q_1, Q_2) *where* $P_1 = Q_1$ *and* $P_2 = Q_2$*.*
- *If* P *is of the form* $\{P'\}_K$*, then* $P = Q$ *iff* Q *is of the form* $\{Q'\}_K$ *where* $P' = Q'$*.*

(Note that we call these 'classical' patterns. This is to distinguish them from the more complex patterns that we will consider later in this paper.)

One last complication remains before we can define formal equivalence. The first thing coming to mind is to say that two expressions are equivalent if their patterns are equal. However, consider two very simple formal expressions K_1 and K_2. Then these formal expressions would not be equivalent. On the other hand, these two expressions have the same meaning: a randomly drawn key. Despite being given different names, they both represent samples from the same distribution. It does not matter if we replace one of them with the other. More generally, we wish to formalize the notion of equivalence in such a way that renaming the keys yields in equivalent expression. Therefore, two formal expressions should be equivalent if their patterns differ only in the names of their keys.

Definition 6 (Key-Renaming Function). *A bijection* σ : ***Keys*** $\rightarrow$ ***Keys*** *is called a* key-renaming function*. For any expression (or pattern)* M*,* $M\sigma$ *denotes the expression (or pattern) obtained from* M *by replacing all occurrences of keys* K *in* M *by* $\sigma(K)$*.*

We are finally able to formalize the symbolic notion of equivalence:

Definition 7 (Equivalence of Expressions). *We say that two expressions M and N are* equivalent*, denoted by $M \cong N$, if there exists a key-renaming function σ such that pattern$(M) =$ pattern$(N\sigma)$.*

3.3 The Computational Model

The fundamental objects of the computational world are strings, $\mathbf{strings} = \{0,1\}^*$, and families of probability distributions over strings. These families are indexed by a *security parameter* $\eta \in \mathbf{parameters} = \mathbb{N}$ (which can be roughly understood as key-lengths). Two distribution families $\{D_\eta\}_{\eta\in\mathbb{N}}$ and $\{D'_\eta\}_{\eta\in\mathbb{N}}$ are *indistinguishable* [30, 54] if no efficient algorithm can determine from which distribution a value was sampled, except with negligible probability:

Definition 8 (Negligible Function). *A function $f : \mathbb{N} \to \mathbb{R}$ is said to be* negligible*, written $f(n) \leq \operatorname{neg}(n)$, if for any $c > 0$ there is an $n_c \in \mathbb{N}$ such that $f(n) \leq n^{-c}$ whenever $n \geq n_c$.*

Definition 9 (Indistinguishability). *Two families $\{D_\eta\}_{\eta\in\mathbb{N}}$ and $\{D'_\eta\}_{\eta\in\mathbb{N}}$, are* indistinguishable*, written $D_\eta \approx D'_\eta$, if for all PPT adversaries* A*,*

$$\left|\Pr\left[d \longleftarrow D_\eta; \mathsf{A}(1^\eta, d) = 1\right] - \Pr\left[d \longleftarrow D'_\eta; \mathsf{A}(1^\eta, d) = 1\right]\right| \leq \operatorname{neg}(\eta)$$

In this model, pairing is an injective *pairing function* $[\cdot,\cdot] : \mathbf{strings} \times \mathbf{strings} \to \mathbf{strings}$ such that the length of the result only depends on the length of the paired strings. An encryption scheme (formalized in the notation of [17]) is a triple of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with key generation $\mathcal{K}$, encryption $\mathcal{E}$ and decryption $\mathcal{D}$. Let $\mathbf{plaintexts}$, $\mathbf{ciphertexts}$, and $\mathbf{keys}$ be nonempty subsets of $\mathbf{strings}$. The set $\mathbf{coins}$ is some probability field that stands for coin-tossing, *i.e.*randomness.

Definition 10 (Symmetric Encryption Scheme). *A* computational symmetric encryption scheme *is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where*

- $\mathcal{K} : \mathbf{parameters} \times \mathbf{coins} \to \mathbf{keys}$ *is a key-generation algorithm;*
- $\mathcal{E} : \mathbf{keys} \times \mathbf{strings} \times \mathbf{coins} \to \mathbf{ciphertexts}$ *is an encryption function;*
- $\mathcal{D} : \mathbf{keys} \times \mathbf{strings} \to \mathbf{plaintexts}$ *is such that for all $k \in \mathbf{keys}$ and $\omega \in \mathbf{coins}$,*

$$\mathcal{D}(k, \mathcal{E}(k, m, \omega)) = m \text{ for all } m \in \mathbf{plaintexts},$$
$$\mathcal{D}(k, \mathcal{E}(k, m', \omega)) = \perp \text{ for all } m' \notin \mathbf{plaintexts}.$$

All of $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ are computable in polynomial-time in the length of the security parameter. When referring to $\mathcal{K}$ and $\mathcal{E}$ algorithms we often omit the argument corresponding to $\mathbf{coins}$. We use the notation $k \longleftarrow \mathcal{K}(1^\eta)$, respectively $y \longleftarrow \mathcal{E}(k, x)$, to denote the generation of a key, respectively a ciphertext, using a uniform source of randomness.

This definition, note, does not include any notion of security, and this must be defined separately. In fact, there are several different such definitions. Abadi and Rogaway [2, 3] consider a spectrum of notions of their own devising, from ‘type-0’ to ‘type-7.’ Their main result uses the strongest of these notions, type-0:

Definition 11 (Type-0 Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. We say that the encryption-scheme is* type-0 secure *if no PPT adversary* A *can distinguish the pair of oracles $(\mathcal{E}(k,\cdot), \mathcal{E}(k',\cdot))$ and $(\mathcal{E}(k,0), \mathcal{E}(k,0))$ as k and k' are randomly generated, that is, for all PPT adversaries* A*:*

$$\Pr[k, k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot),\mathcal{E}(k',\cdot)}(1^\eta) = 1] - \Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,0),\mathcal{E}(k,0)}(1^\eta) = 1] \leq \mathrm{neg}(\eta)\,.$$

Intuitively the above formula says the following: The adversary is given one of two pairs of oracles to interact with, either $(\mathcal{E}(k,\cdot), \mathcal{E}(k',\cdot))$ or $(\mathcal{E}(k,0), \mathcal{E}(k,0))$ (where the keys were randomly generated prior to handing the pair to the adversary), but it does not know which. Then, the adversary can perform any (probabilistic polynomial-time) computation, including several queries to the oracles. It can even query the oracles with messages that depend on previously given answers of the oracles. (The keys used by the oracles for encryption do not change while the adversary queries the oracles.) After this game, the adversary has to decide with which pair of oracles it was interacting. The adversary wins the game if he can decide for the correct one with a probability non-negligibly bigger than $\frac{1}{2}$, or (equivalently) if it can distinguish between the two. If this difference is negligible, as a function of η, we say the encryption scheme is type-0 secure.

As Abadi and Rogaway show, type-0 security is strong enough to provide *soundness* to the formal model. But to see this, we must first explain how the two models can be related.

3.4 The Interpretation Function, Soundness and Completeness

In order to prove any relationship between the formal and computational worlds, we need to define the *interpretation* of expressions and patterns. Once an encryption scheme is picked, we can define the interpretation function Φ, which assigns to each expression or pattern M a family of random variables $\{\Phi_\eta(M)\}_{\eta\in\mathbb{N}}$ such that each $\Phi_\eta(M)$ takes values in $\mathbf{strings}$. As in Abadi and Rogaway [3], this interpretation is defined in an algorithmic way in Figure 3.4. Intuitively,

- Blocks are interpreted as $\mathbf{strings}$,
- Each key is interpreted by running the key generation algorithm,
- Pairs are translated into computational pairs,
- Formal encryptions terms are interpreted by running the (probabilistic) encryption algorithm on the interpretation of the plaintext and the interpretation of the key.

For an expression M, we will denote by $[\![M]\!]_{\Phi_\eta}$ the distribution of $\Phi_\eta(M)$ and by $[\![M]\!]_\Phi$ the ensemble of $\{[\![M]\!]_{\Phi_\eta}\}_{\eta\in\mathbb{N}}$.

Then soundness and completeness are defined in the following way:

Definition 12 (Soundness (Classical)). *We say that an interpretation is* sound in the classical sense*, or that an encryption scheme* provides classical soundness*, if the interpretation Φ (resulting from the encryption scheme) is such that for any given pairs of expressions M and N*

$$M \cong N \Rightarrow [\![M]\!]_\Phi \approx [\![N]\!]_\Phi.$$

The primary result of Abadi and Rogaway given in [3] is that type-0 security provides classical soundness if the expressions M and N have no key cycles.

Soundness has a counterpart, completeness. One can consider soundness to be the property that formal indistinguishability always becomes computational indistinguishability. One can think of completeness as the converse: computational indistinguishability is always the result of formal indistinguishability:

Definition 13 (Completeness (Classical)). *We say that an interpretation is* complete *(in the classical sense), or that an encryption scheme* provides (classical) completeness*, if the interpretation Φ (resulting from the encryption scheme) is such that*

$$[\![M]\!]_{\Phi} \approx [\![N]\!]_{\Phi} \Rightarrow M \cong N$$

for any expressions M and N.

For the proof of the soundness result, it was convenient for Abadi and Rogaway to introduce the interpretation of any pattern M (although this is not absolutely necessary). Therefore, we define interpretation of boxes as follows:

- $\square$ is interpreted by running the encryption algorithm on the fixed plaintext 0 and a randomly generated key.

The precise definition of $\Phi_\eta(P)$ for any pattern P is given by the algorithms in Figure 1. The random variable $\Phi_\eta(P)$ is defined as $\mathrm{INITIALIZE}(1^\eta, P)$ followed by $\mathrm{CONVERT}(P)$. This is a random variable that has values in $\mathbf{strings}$.

We note that these algorithms are fully defined for patterns, and because the grammar for patterns contains the grammar for expressions as a sub-grammar, they are fully defined for expressions as well.

4 Soundness in the Presence of Key Cycles

In this section, we will address the problem of soundness of formal encryption in the presence of key cycles. Later we will see that key cycles do not pose any problem for completeness.

As discussed in the introduction, previous soundness results cannot be applied to messages that contain key cycles. One can immediately ask if there is some impossibility result regarding key cycles, or if this is just a problem in the way proof is conducted.

We start this section by showing that, soundness in the presence of key cycles is not possible to prove with the security notion adopted by Abadi and Rogaway. We suggest a new notion of security, KDM-security [19] as a solution for the problem. In order to prove soundness, we will also need to extend our formal model, and after that we conclude this section showing that with this new definition of security it is possible to obtain soundness even in the presence of key cycles.

4.1 Type-0 Security is Not Enough

In this section we show that type-0 security is not strong enough to ensure soundness in the case of key cycles. That is, we demonstrate that it is possible to construct encryption schemes that are type-0, but fail to provide soundness in the presence of key cycles.

algorithm $\mathrm{INITIALIZE}(1^\eta, P)$
  for $K \in \mathit{Keys}(P)$ **do** $\tau(K) \longleftarrow \mathcal{K}(1^\eta)$
  let $k_0 \longleftarrow \mathcal{K}(1^\eta)$

algorithm $\mathrm{CONVERT}(P)$
  if $P = K$ where $K \in$ **Keys then**
    return $\tau(K)$
  if $P = B$ where $B \in$ **Blocks then**
    return B
  if $P = (P_1, P_2)$ **then**
    $x \longleftarrow \mathrm{CONVERT}(P_1)$
    $y \longleftarrow \mathrm{CONVERT}(P_2)$
    return $[x, y]$
  if $P = \{P_1\}_K$ **then**
    $x \longleftarrow \mathrm{CONVERT}(P_1)$
    $y \longleftarrow \mathcal{E}(\tau(K), x)$
    return y
  if $P = \Box$, **then**
    $y \longleftarrow \mathcal{E}(k_0, 0)$
    return y

Fig. 1. Algorithmic components of the interpretation function

Theorem 14. *Type-0 security does not imply soundness of messages with key cycles. That is, if there exists an encryption scheme that is type-0 secure, then there exists another encryption scheme which is also type-0 secure but does not provide soundness for messages with key cycles.*

Proof. This is shown via a simple counter-example. Assuming that there exists a type-0 secure encryption scheme, we will use it to construct another scheme which is also type-0 secure. However, we will show that this new scheme allows the adversary to distinguish one particular expression M from another particular expression N, even though $M \cong N$.

Let M be $(\{K\}_K, \{000\}_K)$ and let N be the expression $(\{K_1\}_{K_2}, \{000\}_{K_2})$. Since these two expressions are equivalent, an encryption scheme that enforces soundness requires that the family of distributions:

$$\{k \longleftarrow \mathcal{K}(1^\eta); c_1 \longleftarrow \mathcal{E}(k, k); c_2 \longleftarrow \mathcal{E}(k, 000) : [c_1, c_2]\}_{\eta \in \mathbb{N}}$$

be indistinguishable from the family of distributions:

$$\{k_1, k_2 \longleftarrow \mathcal{K}(1^\eta); c_1 \longleftarrow \mathcal{E}(k_2, k_1); c_2 \longleftarrow \mathcal{E}(k_2, 000) : [c_1, c_2]\}_{\eta \in \mathbb{N}}$$

However, this is not implied by Definition 11. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a type-0 secure encryption scheme. Then, using Π, we construct a second type-0 secure encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows:

- Let $\mathcal{K}' = \mathcal{K}$,

– Let $\mathcal{E}'$ be the following algorithm:

$$\mathcal{E}'(k, m, \omega) = \begin{cases} k & \text{if } m = k \\ \mathcal{E}(k, k, \omega) & \text{if } m \neq k \text{ and } \mathcal{E}(k, m, \omega) = k \\ \mathcal{E}(k, m, \omega) & \text{otherwise} \end{cases}$$

– Let $\mathcal{D}'$ be the following algorithm:

$$\mathcal{D}'(k, c) = \begin{cases} k & \text{if } c = k \\ \mathcal{D}(k, k) & \text{if } \mathcal{D}(k, c) = k \\ \mathcal{D}(k, c) & \text{otherwise} \end{cases}$$

It is immediate that $\mathcal{E}$ is *PPT*, and Π' is type-0 secure. To see this, suppose that Π' is not type-0 secure. As it differs from they type-0 secure Π only on k and an (m, ω) pair such that $\mathcal{E}(k, m, \omega) = k$, this implies that the adversary could successfully guess k. However, that would contradict the type-0 security of Π.

Thus, the new scheme Π' must also be type-0 secure. However, it does not guarantee indistinguishability for the two distributions above. The first distribution will always output the key k paired with the encryption of 000 with k, while the second outputs two ciphertexts. An adversary may easily distinguish the two by using the first term of the pair to decrypt the second and comparing the decryption with 000. □

Remark 15. We note that in the proof, the expression M contains a key cycle of length 1. What if all key cycles are of length 2 or more? This question remains open. That is, there is no known type-0 secure encryption scheme which fails to provide soundness for key cycles that are of length two or more.

Because type-0 encryption implies types 1 through 7, Theorem 14 implies that soundness with key cycles cannot be provided by the security definitions devised by Abadi and Rogaway. In the next section, we show that this soundness property can, however, be met with *new* computational definitions.

4.2 KDM-Security

In the last section, we showed that the notions of security found in [2, 3] are not strong enough to enforce soundness in the presence of key cycles. However, *key-dependent message* (KDM) security, which was introduced by Black *et al.* [19] (and in a weaker form by Camenisch and Lysyanskaya [20]), is strong enough to enforce soundness even in this case. (We note that Camenisch and Lysyanskaya also provided a natural application of KDM security, a credential system with interesting revocation properties, and so KDM security is of independent interest as well.)

KDM security both strengthens and weakens type-0 security. Recall that type-0 security allows the adversary to submit messages to an oracle which does one of two things:

– It could encrypt the message twice, under two different keys, or
– It could encrypt the bit 0 twice, under the same key.

An encryption scheme is type-0 secure if no adversary can tell which of these is being done. For KDM security, however, the game is slightly different. To over-simplify:

– The oracle in the KDM-security encrypts once, under one single key.
– Further, it encrypts either the message, or a *string* of 0's of equivalent length.
– However, it is willing to encrypt not just messages from the adversary, but also (more generally) *functions of the secret key*.

The first two of these differences make KDM security weaker than type-0 security. Specifically type-0 security conceals both the length of the plaintext and whether two ciphertexts were created using the same key or with two different ones. KDM security does not necessarily conceal either of these things. The last difference, however, is a significant strengthening. As its name suggests, KDM security remains strong even when the messages depend on the secret key—which, as Theorem 14 shows, is not necessarily true for type-0 security.

To provide the full picture, KDM security is defined in terms of *vectors* of keys and functions over these vectors. It is also defined in terms of oracles $\mathsf{Real}_{\bar{\mathbf{k}}}$ and $\mathsf{Fake}_{\bar{\mathbf{k}}}$, which work as follows: suppose that for a fixed security parameter $\eta \in \mathbb{N}$, a vector of keys is given: $\bar{\mathbf{k}} = \{k_i\}_{i\in\mathbb{N}}$ with $k_i \longleftarrow \mathcal{K}(1^\eta)$. (In each run of the key-generation algorithm independent coins are used.) The adversary can now query the oracles providing them with a pair (j, g), where $j \in \mathbb{N}$ and $g : \mathbf{keys}^\infty \to \{0,1\}^*$ is a constant length, deterministic function:

– The oracle $\mathsf{Real}_{\bar{\mathbf{k}}}$ when receiving this input returns $c \longleftarrow \mathcal{E}(k_j, g(\bar{\mathbf{k}}))$;
– The oracle $\mathsf{Fake}_{\bar{\mathbf{k}}}$ when receiving this same input returns $c \longleftarrow \mathcal{E}(k_j, 0^{|g(\bar{\mathbf{k}})|})$.

The challenge facing the adversary is to decide whether it has interacted with oracle $\mathsf{Real}_{\bar{\mathbf{k}}}$ or oracle $\mathsf{Fake}_{\bar{\mathbf{k}}}$. Formally:

Definition 16 (Symmetric-KDM Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let the two oracles $\mathsf{Real}_{\bar{\mathbf{k}}}$ and $\mathsf{Fake}_{\bar{\mathbf{k}}}$ be as defined above. We say that the encryption scheme is* (symmetric) KDM-secure *if for all PPT adversaries* A*:*

$$\Pr\left[\bar{\mathbf{k}} \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathsf{Real}_{\bar{\mathbf{k}}}}(1^\eta) = 1\right] - \Pr\left[\bar{\mathbf{k}} \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathsf{Fake}_{\bar{\mathbf{k}}}}(1^\eta) = 1\right] \le \operatorname{neg}(\eta)$$

An implementation of this definition in the random oracle model is provided by Black *et al.* [19]. If RO is a random oracle (*i.e.*, a randomly-chosen function from $\{0,1\}^*$ to $\{0,1\}^\infty$) and by $\langle \cdot, \cdots, \cdot \rangle$ we mean the concatenation of the strings, then the scheme of Black *et al.* is the triple of algorithms $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ where

– $\mathcal{K}'(1^\eta)$: select and return $k_1 \longleftarrow \{0,1\}^\eta$.
– $\mathcal{E}'(k_1, m)$: select $r \longleftarrow \{0,1\}^\eta$; let $y := m \oplus RO(\langle k_1, r\rangle)$ (where only the first $|m|$ bits of the oracle's output are used in the XOR); return $\langle y, r\rangle$.
– $\mathcal{D}'(k_1, c = \langle y, r\rangle)$: let $x := y \oplus RO(\langle k_1, r\rangle)$; return x.

In this work, we will consider a minor variant of this scheme, modified so as to be *strictly which-key revealing*—a property we will consider in our discussion of completeness (Section 5.6).[6] Our scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is the triple of algorithms:

[6] Another small difference is that the formalism of Black *et al.* assumes that symmetric encryption schemes operate only on plaintexts of a fixed, given size, while we use the more general

- $\mathcal{K}(1^\eta)$: select $k_1 \longleftarrow \{0,1\}^\eta$ and $k_2 \longleftarrow \{0,1\}^\eta$; return $k = \langle k_1, k_2 \rangle$.
- $\mathcal{E}\,(k = \langle k_1, k_2 \rangle, m)$: select $r \longleftarrow \{0,1\}^\eta$; let $y := m \oplus RO(\langle k_1, r \rangle)$; return $\langle y, r, k_2 \rangle$.
- $\mathcal{D}(k = \langle k_1, k_2 \rangle, c = \langle y, r, k' \rangle)$: check that $k' = k_2$; if so, let $x := y \oplus RO(\langle k_1, r \rangle)$; return x.

The scheme Π is exactly the KDM-secure scheme Π', except that the key now has a second component k_2 which is not used to encrypt but is appended to the ciphertext. (Again, this seemingly-extraneous component will become important when we discuss completeness in Section 5.6.)

We quickly prove that this addition does not invalidate the KDM-security of the scheme: If there exists an adversary A which can distinguish the real oracle from the fake oracle when the oracle uses our scheme, then there exists a second adversary A' which can do the same for the scheme of Black *et al.* To see this, let A' simulate A. First A' generates as many keys $k_{2,j}$ $j = 1, ..., n$ as the maximum number of encrypting keys that are used in A's queries of the form $\langle j, g \rangle$. When A' constructs the queries$\langle j, g \rangle$ for oracle (using the original Black *et al.* encryption), it appends to each encryption in g that is requested from the oracle the corresponding string $k_{2,i}$. When A' passes the query $\langle j, g \rangle$ to the oracle, and get a response of the form $\langle y, r \rangle$, it returns $\langle y, r, k_{2,j} \rangle$ to A. In this way, A' exactly simulates the adversary A, and so A' will successfully attack the scheme of Black *et al.* with at least the same probability with which A successfully attacks Π. But because the scheme of Black *et al.* is KDM-secure, this probability must be negligible. Hence, the probability that A can successfully attack our scheme must be negligible as well.

Remark 17. Both the original scheme by Black *et al.* and the modified form of it above use the random oracle. In fact, all known implementations of KDM-security exist in the random-oracle model. However, we note that the general definition of KDM-security is well-founded even in the standard model. We also note that this definition is phrased in terms of indistinguishability. While one could also imagine analogous definitions phrased in terms of non-malleability, an exploration of those definitions is beyond the scope of this paper.

We note that KDM-security implies type-3 security:

Definition 18 (Type-3 Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. We say that the encryption-scheme is* type-3 secure *if no PPT adversary* A *can distinguish the oracles $\mathcal{E}(k,\cdot)$ and $\mathcal{E}(k, 0^{|\cdot|})$ as k is randomly generated, that is, for all PPT adversaries* A*:*

$$\Pr\Big[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot)}(1^\eta) = 1\Big] - \Pr\Big[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,0^{|\cdot|})}(1^\eta) = 1\Big] \leq \mathrm{neg}\,(\eta)$$

In fact, the definition of type-3 encryption is exactly the same as that for KDM-security, except that the adversary must submit concrete messages to the encryption oracle instead of functions. But since the functions submitted in KDM security can be the constant function that always produce a single output, the type-3 security 'game' is a special case of that for KDM security.

---

definition which allows variable-length plaintexts. For this definition, however, the difference is moot: the function g in Definition 16 always produces output of a fixed length.

On the other hand, KDM security does not attempt to conceal that two ciphertexts were created with the same key (type-1 security) nor the length of the plaintext (type-2 security). It will therefore be impossible for KDM security to provide soundness in the classical sense (Definition 12). Nonetheless, a weaker form of soundness can be achieved if the formal model is also slightly weakened.

4.3 Weakening the Symbolic Model

In this section, we develop a weaker version of the formal model—one that allows formal encryption to leak partial information about the plaintext and the key. One can think of this as a preview or a special case of Section 5, where we discuss such weakening in general. In this section, however, we focus on the partial leakage allowed (in the computational model) by KDM security: the length of the plaintext, and whether two different ciphertexts were created using the same key.

To model the leakage of plaintext length, we first need to add the very concept of 'length' to the formal model:

Definition 19 (Formal Length). *A formal length-function is a function symbol with fresh letter ℓ satisfying at least the following identities:*

- *For all blocks B_1 and B_2, $\ell(B_1) = \ell(B_2)$ iff $|B_1| = |B_2|$,*
- *For all keys K and K', $\ell(K) = \ell(K')$,*
- *If $\ell(M_1) = \ell(N_1)$, $\ell(M_2) = \ell(N_2)$ then $\ell((M_1, M_2)) = \ell((N_1, N_2))$,*
- *If $\ell(M) = \ell(N)$, then for all K, $\ell(\{M\}_K) = \ell(\{N\}_K)$, and*
- *For all keys K and K', $\ell(\{M\}_K) = \ell(\{M\}_{K'})$.*

We would like to emphasize that these are the identities that a formal length function *minimally* has to satisfy. There may be more. In fact, if we only assume these properties, there is no hope to obtain completeness. We also remark, that it follows that for any key-renaming function σ, and expression M, $\ell(M) = \ell(M\sigma)$.

Given this, it is straightforward to add the required leakage to the formal model. If patterns represents those aspects of an expression that can be learned by the adversary, then patterns must now reveal the plaintext-length and key-names for undecryptable terms:

Definition 20 (Pattern (Type-3)). *We define the* set of patterns*, **Pat**, by the grammar:*

$$\textbf{\textit{Pat}} ::= \textbf{\textit{Keys}} \mid \textbf{\textit{Blocks}} \mid (\textbf{\textit{Pat}}, \textbf{\textit{Pat}}) \mid \{\textbf{\textit{Pat}}\}_{\textbf{\textit{Keys}}} \mid \Box_{\textbf{\textit{Keys}}, \ell(\textbf{\textit{Exp}})}$$

The type-3 pattern of an expression M, denoted by $pattern_3(M)$, is derived from M by replacing each encryption term $\{M'\}_K \in vis(M)$ (where $K \notin$ R-Keys(M)) by $\Box_{K,\ell(M')}$.

Note that the only difference between a type-3 pattern and a classical pattern is that an undecryptable term $\{M\}_K$ becomes $\Box_{K,\ell(M)}$ (*i.e.*labeled with the key and length) in type-3 patterns instead of merely $\Box$ in classical patterns.

Our notion of formal equality must be updated as well.

Definition 21 (Formal Equivalence (Type-3)). *For two patterns P and Q, $P =_3 Q$ is defined in the following way:*

- *If $P \in$ **Blocks** $\cup$ **Keys**, then $P =_3 Q$ iff P and Q are identical.*
- *If P is of the form $\square_{K,\ell(M')}$, then $P =_3 Q$ iff Q is of the form $\square_{K,\ell(N')}$, and $\ell(M') = \ell(N')$ in the sense of Definition 19.*
- *If P is of the form (P_1, P_2), then $P =_3 Q$ iff Q is of the form (Q_1, Q_2) where $P_1 =_3 Q_1$ and $P_2 =_3 Q_2$.*
- *If P is of the form $\{P'\}_K$, then $P =_3 Q$ iff Q is of the form $\{Q'\}_K$ where $P' =_3 Q'$.*

We say that expressions M and N are equivalent in the type-3 sense*, denoted by $M \cong_3 N$, if there exists a key-renaming function σ such that pattern$_3(M) =_3$ pattern$_3(N\sigma)$. (Since a key-renaming function replaces all occurrences of K with $\sigma(K)$, we note that under σ, $\square_{K,\ell(M)}$ will become $\square_{\sigma(K),\ell(M\sigma)}$.)*

Lastly, the above change to formal equivalence requires that the notions of soundness and completeness be similarly altered:

Definition 22 (Soundness (Type-3)). *We say that an interpretation is* type-3 sound*, or that an encryption scheme* provides soundness in the type-3 sense*, if the interpretation Φ (resulting from the encryption scheme) is such that*

$$M \cong_3 N \Rightarrow [\![M]\!]_\Phi \approx [\![N]\!]_\Phi .$$

for any pair of expressions M and N.

Definition 23 (Completeness (Type-3)). *We say that an interpretation is* type-3 complete*, or that an encryption scheme* provides completeness in the type-3 sense*, if the interpretation Φ (resulting from the encryption scheme) is such that for any pair of expressions M and N,*

$$[\![M]\!]_\Phi \approx [\![N]\!]_\Phi \Rightarrow M \cong_3 N.$$

4.4 Soundness for Key Cycles

Below, we present our main soundness result for key cycles: if an encryption scheme is KDM secure, it also provides type-3 soundness even in the presence of key cycles. We then show that type-0 security does not imply, nor is implied by KDM security.

Proposition 24. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a computational symmetric encryption scheme such that for each η, if $k, k' \longleftarrow \mathcal{K}(1^\eta)$, then $|k| = |k'|$, and for each m plaintext, $|\mathcal{E}(k, m, w)| = |\mathcal{E}(k', m, w')|$ for all $w, w' \longleftarrow$ $\mathbf{coins}$. If $\ell(M) = \ell(N)$ and the length-function ℓ satisfies the equalities listed in Definition 19, then $|\Phi_\eta(M)| = |\Phi_\eta(N)|$.*

Proof. Observe, that if $|m| = |m'|$, then $|\mathcal{E}(k, m, w)| = |\mathcal{E}(k', m', w')|$ because of type-3 security. The rest is straightforward from Definition 19.

Theorem 25 (Symmetric KDM Security Implies Soundness). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a computational symmetric encryption scheme such that for each η, if $k, k' \longleftarrow \mathcal{K}(1^\eta)$, then $|k| = |k'|$, and for each m plaintext, $|\mathcal{E}(k, m, w)| = |\mathcal{E}(k', m, w')|$ for all $w, w' \longleftarrow$ $\mathbf{coins}$.*

If Π is KDM-secure and the length-function ℓ satisfies only the equalities listed in Definition 19, then Π provides type-3 soundness.

Proof. We first redefine the interpretation of patterns. The only thing we have to change in the interpretation of Abadi and Rogaway is the interpretation of a box. Now, we interpret a pattern $\Box_{K,\ell(M)}$ for a given security parameter η as $\Phi_\eta(\{0^{|\Phi_\eta(M)|}\}_K)$. That is, the interpretation function (that used to encrypt a single 0 under a random key) now encrypts a string of 0's of the same requisite length (length of $\Phi_\eta(M)$) under the correct key $\tau(K)$.

The proof in this case is a somewhat reduced hybrid argument. In a standard hybrid argument, like the one that Abadi and Rogaway used to prove their soundness result, several patterns are put between M and N; then, using security, it is proven that soundness holds between each two consecutive patterns, and therefore soundness holds for M and N. In our case, we first directly prove that $[\![M]\!]_\Phi$ is indistinguishable from $[\![\textit{pattern}_3(M)]\!]_\Phi$. Then, since that holds for N too, and since $\textit{pattern}_3(M)$ differs from $\textit{pattern}_3(N)$ only in the name of keys, $[\![\textit{pattern}_3(M)]\!]_\Phi$ is indistinguishable from $[\![\textit{pattern}_3(N)]\!]_\Phi$, therefore the result follows. KDM security is used to show that $[\![M]\!]_\Phi$ and $[\![\textit{pattern}_3(M)]\!]_\Phi$ are indistinguishable.

For an arbitrary (formal) key K, let $\iota(K)$ denote the index of K. For an expression M, a set of formal (unrecoverable) keys S, and a function $\tau : \textbf{Keys} \setminus S \to \mathbf{keys}$, we define a function $f_{M,S,\tau} : \mathbf{coins}^{e(M)} \times \mathbf{keys}^\infty \to \mathbf{strings}$ (where $e(M)$ is the number of encryptions in M) recursively in the following way (k_i is the i'th component of $\bar{\mathbf{k}}$):

- For $M = B \in \textbf{Blocks}$, let $f_{B,S,\tau} : \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as $f_{B,S,\tau}(\bar{\mathbf{k}}) = B$;
- For $M = K \in \textbf{Keys} \cap S$, let $f_{K,S,\tau} : \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as $f_{K,S,\tau}(\bar{\mathbf{k}}) = k_{\iota(K)}$;
- For $M = K \in \textbf{Keys} \setminus S$, let $f_{K,S,\tau} : \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as $f_{K,S,\tau}(\bar{\mathbf{k}}) = \tau(K)$;
- For $M = (M_1, M_2)$, let $f_{(M_1,M_2),S,\tau} : \mathbf{coins}^{e(M_1)} \times \mathbf{coins}^{e(M_2)} \times \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as the computational pairing $f_{(M_1,M_2),S,\tau}(\omega_{M_1}, \omega_{M_2}, \bar{\mathbf{k}}) = [f_{M_1,S,\tau}(\omega_{M_1}, \bar{\mathbf{k}}), f_{M_2,S,\tau}(\omega_{M_2}, \bar{\mathbf{k}})]$;
- For $M = \{N\}_K$ and $K \in S$, let $f_{\{N\}_K,S,\tau} : \mathbf{coins} \times \mathbf{coins}^{e(N)} \times \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as $f_{\{N\}_K,S,\tau}(\omega, \omega_N, \bar{\mathbf{k}}) = \mathcal{E}(k_{\iota(K)}, f_{N,S,\tau}(\omega_N, \bar{\mathbf{k}}), \omega)$;
- For $M = \{N\}_K$ and $K \notin S$, let $f_{\{N\}_K,S,\tau} : \mathbf{coins} \times \mathbf{coins}^{e(N)} \times \mathbf{keys}^\infty \to \mathbf{strings}$ be defined as $f_{\{N\}_K,S,\tau}(\omega, \omega_N, \bar{\mathbf{k}}) = \mathcal{E}(\tau(K), f_{N,S,\tau}(\omega_N, \bar{\mathbf{k}}), \omega)$.

We note that this function is constant length because according to our assumptions, keys are constant-length (for the same η) and the length of an encryption only depends on the length of the message and η. We first prove that $[\![M]\!]_\Phi \approx [\![\textit{pattern}_3(M)]\!]_\Phi$. Suppose that $[\![M]\!]_\Phi \not\approx [\![\textit{pattern}_3(M)]\!]_\Phi$. This means that there is an adversary A that distinguishes the two distributions, that is

$$\Pr\left[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1\right] - \Pr\left[x \longleftarrow [\![\textit{pattern}_3(M)]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1\right]$$

is a non-negligible function of η. We will show that this contradicts the fact that the system is (symmetric) KDM-secure. To this end, we construct an adversary that can distinguish whether oracle $\mathcal{F}$ is $\mathsf{Real}_{\bar{\mathbf{k}}}$ or $\mathsf{Fake}_{\bar{\mathbf{k}}}$. From now on, let $S = \textbf{Keys} \setminus \textit{R-Keys}(M)$. Consider the following algorithm:

algorithm $\mathsf{B}^{\mathcal{F}}(1^\eta, M)$
  for $K \in$ *R-Keys*(M) **do** $\tau(K) \longleftarrow \mathcal{K}(1^\eta)$
  $y \longleftarrow \text{CONVERT2}(M, M)$
  $b \longleftarrow \mathsf{A}(1^\eta, y)$
  return b

algorithm CONVERT2(M', M) with $M' \sqsubseteq M$
  if $M' = K$ where $K \in$ *R-Keys*(M) **then**
    return $\tau(K)$
  if $M' = B$ where $B \in$ **Blocks then**
    return B
  if $M' = (M_1, M_2)$ **then**
    $x \longleftarrow \text{CONVERT2}(M_1, M)$
    $y \longleftarrow \text{CONVERT2}(M_2, M)$
    return $[x, y]$
  if $M' = \{M_1\}_K$ with $K \in$ *R-Keys*(M) **then**
    $x \longleftarrow \text{CONVERT2}(M_1, M)$
    $y \longleftarrow \mathcal{E}(\tau(K), x)$
    return y
  if $M' = \{M_1\}_K$ with $K \notin$ *R-Keys*(M) **then**
    $\omega \longleftarrow \mathbf{coins}^{e(M_1)}$
    $y \longleftarrow \mathcal{F}(\iota(K), f_{M_1,S,\tau}(\omega, .))$
    return y

This algorithm applies the distinguisher $\mathsf{A}(1^\eta, \cdot)$ to distribution $[\![M]\!]_\Phi$ when $\mathcal{F}$ is $\mathsf{Real}_{\bar{\mathbf{k}}}$, and to distribution of $[\![\textit{pattern}_3(M)]\!]_\Phi$ when $\mathcal{F}$ is $\mathsf{Fake}_{\bar{\mathbf{k}}}$. If $\mathsf{A}(1^\eta, \cdot)$ can distinguish $[\![M]\!]_\Phi$ and $[\![\textit{pattern}_3(M)]\!]_\Phi$, then $\mathsf{B}^{\mathcal{F}}(1^\eta, \cdot)$ can distinguish $\mathsf{Real}_{\bar{\mathbf{k}}}$ and $\mathsf{Fake}_{\bar{\mathbf{k}}}$ —a contradiction. Hence $[\![M]\!]_\Phi \approx [\![\textit{pattern}_3(M)]\!]_\Phi$.

In a similar manner, we can show that $[\![N]\!]_\Phi \approx [\![\textit{pattern}_3(N)]\!]_\Phi$. Finally, it is easy to see that $[\![\textit{pattern}_3(M)]\!]_\Phi = [\![\textit{pattern}_3(N)]\!]_\Phi$, because the two patterns differ only by key renaming. Hence $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$. $\square$

We conclude our consideration of KDM security by demonstrating what Black *et al.* claimed informally: the notion of KDM security is 'orthogonal' to the previous definitions of security. In particular, we show that KDM security neither implies nor is implied by type-0 security.

Proposition 26. *Type-0 security does not imply (symmetric) KDM-security. If there exists an encryption scheme that is type-0 secure, there exists an encryption scheme which is also type-0 secure but not KDM-secure.*

Proof. Suppose that there exists a type-0 secure encryption scheme. In the proof of Theorem 14, we constructed a type-0 secure scheme Π' such that with Π', the interpretations of $(\{K\}_K, \{000\}_K)$ and $(\{K_1\}_{K_2}, \{000\}_{K_2})$ are distinguishable. If all type-0 encryptions schemes are KDM-secure, then Π' is as well. However, by the same method as in the proof of Theorem 25, this would mean that with Π', the interpretation of $(\{K\}_K, \{000\}_K)$ and of $(\{K_1\}_{K_2}, \{000\}_{K_2})$ are both indistinguishable from the distribution of $[\mathcal{E}(k, 0^{|k'|}, \omega), \mathcal{E}(k, 000, \omega')]$, as $k, k' \longleftarrow \mathcal{K}(1^\eta)$ and $\omega, \omega' \longleftarrow$ **coins**, that is, the interpretations of $(\{K\}_K, \{000\}_K)$ and of $(\{K_1\}_{K_2}, \{000\}_{K_2})$ are indistinguishable—a contradiction. □

Note, that in the above proof, we do not need the restriction about length that we needed in Theorem 25. The reason is, that the plaintexts submitted to the oracles for encryption, namely $(k, 000)$ and $(k_1, 000)$, have the same distribution, and so their lengths have the same distribution as well. Without the assumptions on the length, Theorem 25 would not be true, not because our formal length function is not good enough to cover that case as for example $[k_1, k_2]$ and $[k, k]$ would have different distribution of lengths.

Proposition 27. *KDM security does not imply type-0 security. That is, there is an encryption scheme that is KDM-secure, but not type-0 secure.*

Proof. The encryption scheme described in Section 4.2 is KDM secure, but reveals length and which-key, so it is not type-0. □

5 Soundness and Completeness of Expansions of the AR Logic for Symmetric Encryption

We saw earlier, how to expand the Abadi-Rogaway logic to handle length and which-key revealing (type-3) encryption schemes by indexing the boxes with length and keys. Motivated by this, we now provide a general treatment of soundness and completeness for the Abadi-Rogaway type logics of formal encryptions. We present a general method to handle encryptions that leak some information. We also allow not only computational, but purely probabilistic interpretations as well, and equivalence notions other then computational indistinguishability.

In Subsection 5.1 we present a general probabilistic framework for symmetric encryptions, which includes both the computational and the information-theoretic encryption schemes. Then, in Subsection 5.2, we show a general way to handle partial leakage of information in the formal view. This will be done essentially via an equivalence relation on the set of (symbolic) encryption terms, which is meant to express which encryption terms are (computational) indistinguishable for an adversary. In that section, we also introduce an important notion of this equivalence relation that we call *properness*. This notion is essential, as properness is exactly the property that makes an Abadi-Rogaway type hybrid argument go through. Finally, in the remaining subsections, we present the interpretation, the general soundness and completeness results, and show that soundness and completeness theorems for length-revealing, and which key revealing cryptographic schemes are particular cases of our general results. As a purely probabilistic example, we consider the One-Time Pad, and show soundness and completeness for it as well.

5.1 A General Treatment for Symmetric Encryptions

We provide a general probabilistic framework for symmetric encryption, which contains both the computational and the information-theoretic description as special cases. Keys, plaintexts and ciphertexts are elements of some discrete set $\overline{\mathbf{strings}}$. This is $(\{0,1\}^*)^\infty$ in the case of a computational treatment, and it is $\{0,1\}^*$ for the information-theoretic description. The elements of $(\{0,1\}^*)^\infty$ are sequences in $\{0,1\}^*$, corresponding to a parameterization by the security parameter.

A fixed subset, $\mathbf{plaintext} \subseteq \overline{\mathbf{strings}}$ represents the messages that are allowed to be encrypted. Another subset, $\mathbf{keys} \subseteq \overline{\mathbf{strings}}$ is the possible set encrypting keys that corresponds to the range of the key generation algorithm $\mathcal{K}$. In order to be able to build up longer messages from shorter ones, we assume that an injective *pairing function* is given: $[\ .\ ,\ .\ ] : \overline{\mathbf{strings}} \times \overline{\mathbf{strings}} \to \overline{\mathbf{strings}}$. The range of the pairing function will be called $\mathbf{pairs}$: $\mathbf{pairs} := \mathrm{Ran}_{[\ .\ ,\ .\ ]}$. A symmetric encryption scheme has the following constituents:

Key-Generation. Key-generation is represented by a random variable $\mathcal{K} : \Omega_{\mathcal{K}} \to \mathbf{keys}$, over a discrete probability field $(\Omega_{\mathcal{K}}, \mathrm{Pr}_{\mathcal{K}})$. In a given scheme, more than one key-generation algorithms are allowed.

Encryption. For a given $k \in \mathbf{keys}$, and a given $x \in \mathbf{plaintext}$, $\mathcal{E}(k,x)$ is a random variable over some discrete probability field $(\Omega_{\mathcal{E}}, \mathrm{Pr}_{\mathcal{E}})$. The values of this random variable are in $\overline{\mathbf{strings}}$ and are denoted by $\mathcal{E}(k,x)(\omega)$, whenever $\omega \in \Omega_{\mathcal{E}}$.

Decryption. An encryption must be decryptable, so we assume that for each $k \in \mathbf{keys}$, a function $\mathcal{D} : (k,y) \mapsto \mathcal{D}(k,y)$ is given satisfying $\mathcal{D}\big(k, \mathcal{E}(k,x)(\omega)\big) = x$ for all $\omega \in \Omega_{\mathcal{E}}$ and $x \in \mathbf{plaintext}$.

If any of these operations are given (as input) an element that is not in the domain, then an error message $\perp$ is returned.

Indistinguishability. The notion of *indistinguishability* is important both in case of computational and information-theoretic treatments of cryptography. It expresses that there is only very small probability to tell two probability distributions apart.

An equivalence relation called *indistinguishability* is defined on distributions over $\overline{\mathbf{strings}}$. We denote this relation by $\approx$. We say that two random variables taking values in $\overline{\mathbf{strings}}$ are equivalent (indistinguishable) if (and only if) their distributions are equivalent; we use $\approx$ for denoting this equivalence between random variables as well. We require, indistinguishability to be invariant under pairing and its inverse; for $\approx$, we require the followings:

(i) Random variables with the same distribution are indistinguishable;
(ii) For random variables $F : \Omega_F \to \overline{\mathbf{strings}}$ and $G : \Omega_G \to \overline{\mathbf{strings}}$, if $F \approx G$, the following must hold: If π^i denotes the projection onto one of the components of $\overline{\mathbf{strings}} \times \overline{\mathbf{strings}}$, then $\pi^i \circ [\cdot,\cdot]^{-1} \circ F \approx \pi^i \circ [\cdot,\cdot]^{-1} \circ G$ for $i = 1, 2$;
(iii) If $F' : \Omega_F \to \overline{\mathbf{strings}}$, $G' : \Omega_G \to \overline{\mathbf{strings}}$ are also indistinguishable random variables such that F and F' are independent and G and G' are also independent, then $\omega_F \mapsto [F(\omega_F), F'(\omega_F)]$ and $\omega_G \mapsto [G(\omega_G), G'(\omega_G)]$ are indistinguishable random variables; moreover, if $\alpha, \beta : \overline{\mathbf{strings}} \to \overline{\mathbf{strings}}$ are functions that preserve $\approx$ (*i.e.*$\alpha \circ F \approx \alpha \circ G$ and $\beta \circ F \approx \beta \circ G$ whenever $F \approx G$), then $\omega_F \mapsto [(\alpha \circ F)(\omega_F), (\beta \circ F)(\omega_F)]$ and $\omega_G \mapsto [(\alpha \circ G)(\omega_G), (\beta \circ G)(\omega_G)]$ are indistinguishable random variables if $F \approx G$.

Indistinguishability needs to satisfy some further properties under encryption and decryption that we will specify under the definition of encryption schemes below.

Definition 28 (General Encryption Scheme). *An* encryption scheme *is a quadruple $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$ where each $\mathcal{K}_i$ is a key-generation, $\mathcal{E}$ is an encryption, $\mathcal{D}$ decrypts ciphertexts encrypted by $\mathcal{E}$, and $\approx$ is the indistinguishability defined above.*

We require that for any $i, j \in I$, the probability distribution of $\mathcal{K}_i$ be distinguishable from any constant in $\mathbf{strings}$, the distributions of $\mathcal{K}_i$ and of $\mathcal{K}_j$ be distinguishable whenever $i \neq j$, and also the joint distribution (k, k') be distinguishable from the distribution (k, k) if k and k' are independently generated: $k \longleftarrow \mathcal{K}_i$, $k' \longleftarrow \mathcal{K}_i$.

The indistinguishability relation $\approx$, besides satisfying the properties stated before, needs to be such that if F and G are random variables taking values in $\mathbf{strings}$, and $\mathcal{K}_i$ is a key-generation such that the distribution of $[\mathcal{K}_i, F]$ is indistinguishable from the distribution of $[\mathcal{K}_i, G]$, then:

(i) Random variables $(\omega_\mathcal{E}, \omega_\mathcal{K}, \omega) \mapsto \mathcal{E}\big(\mathcal{K}_i(\omega_\mathcal{K}), F(\omega)\big)(\omega_\mathcal{E})$ and $(\omega_\mathcal{E}, \omega_\mathcal{K}, \omega) \mapsto \mathcal{E}\big(\mathcal{K}_i(\omega_\mathcal{K}), G(\omega)\big)(\omega_\mathcal{E})$ are indistinguishable;
(ii) $(\omega_\mathcal{K}, \omega) \mapsto \mathcal{D}\big(\mathcal{K}_i(\omega_\mathcal{K}), F(\omega)\big)$ and $(\omega_\mathcal{K}, \omega) \mapsto \mathcal{D}\big(\mathcal{K}_i(\omega_\mathcal{K}), G(\omega)\big)$ are also indistinguishable random variables.

Here the probability over $\Omega_{\mathcal{K}_i} \times \Omega_F$ is the joint probability of $\mathcal{K}_i$ and F, which are here not necessarily independent. Similarly for G. (Note, that if F and G are not taking plaintext values in (i) or ciphertext values in (ii), then simply error message is returned, this does not jeopardize the definition.)

Example 29 (Computational Indistinguishability and Encryption). The standard notion of computational indistinguishability of [54], Definition 9, is a special case of our general definition of indistinguishability. In this case $\mathbf{strings} = (\{0,1\}^*)^\infty = \mathbf{strings}^\infty$. Random variables of computational interest have the form $F : \Omega_F \to \mathbf{strings}^\infty$ and have independent components; *i.e.*, for a security parameter $\eta \in \mathbb{N}$, denoting by $F_\eta : \Omega_F \to \mathbf{strings}$ the η'th component of F, it is required F_η and $F_{\eta'}$ to be independent random variables whenever $\eta \neq \eta'$. Indistinguishability then is phrased with the ensemble of probability distributions of the components of the random variables. Lastly, the computational encryption as we defined it in Definition 10 is a special case of our general definition of encryption schemes.

The simplest example for indistinguishability is that it holds between two random variables if and only if their distributions are identical. With this indistinguishability notion, we finish this section by presenting a more detailed example for the One-Time Pad.

Example 30 (Information-Theoretical Equivalence and the One-Time Pad). Consider $\mathbf{strings} := \{0,1\}^*$ with the following pairing function: For any two strings $x, y \in \mathbf{strings}$ we can define the pairing of x and y as $[x, y] := \langle x, y, 0, 1^{|y|}\rangle$ The number of 1's at the end indicate how long the second string is in the pair, and the 0 separates the strings from the 1's. Let $\mathbf{blocks}$ be those strings that end with 100. The ending is just a tag, it shows that the string is a block.

Indistinguishability. As we mentioned, let us now call two distributions indistinguishable, if they are identical, and denote this relation by $=_d$.

Key-Generation. In case of the OTP, the length of the encrypting key must match the length of the plaintext. Thus, we need a separate key-generation for each length. That is, for each $n > 3$, $\mathcal{K}_n$ is a random variable over some discrete probability field $(\Omega_{\mathcal{K}_n}, \Pr_{\mathcal{K}_n})$ such that its values are equally distributed over $\mathbf{keys}_n := \{k \mid k \in \mathbf{strings},\ |k| = n,\ k \text{ ends with } 010\}$. Let $\mathbf{keys} := \bigcup_4^\infty \mathbf{keys}_n$. For $k \in \mathbf{keys}$, let $\mathrm{core}(k)$ denote the string that we get from k by cutting the tag 010.

Encryption. Let the domain of the encryption function, $\mathrm{Dom}_\mathcal{E}$, be those elements $(k, x) \in \mathbf{keys} \times \mathbf{strings}$, for which $|k| = |x| + 3$, and let $\mathcal{E}(k, x) := \langle \mathrm{core}(k) \oplus x, 110 \rangle$. The tag 110 informs us that the string is a ciphertext. Notice that this encryption is not probabilistic, hence $\mathcal{E}(k, x)$ is not a random variable (that is, it can be considered as a constant random variable). Notice also, that the tag of the plaintext is not dropped, that part is also encrypted.

Decryption. The decryption function $\mathcal{D}(k, x)$ is defined whenever $|k| = |x|$, and, naturally the value of $\mathcal{D}(k, x)$ is the first $|k| - 3$ bits of $k \oplus x$.

5.2 Equivalence of Expressions

In this section, we will use the notions of *blocks*, *keys*, *expressions*, *subexpressions* introduced in Subsection 3.1.

In their treatment, Abadi and Rogaway defined equivalence of expressions via replacing encryption terms encrypted with non-recoverable keys in an expression by a box; two expressions then were said to be equivalent if once these encryption terms were replaced by the boxes, the obtained *patterns* looked the same up to *key renaming*. This method implicitly assumes, that an adversary cannot distinguish any undecryptable terms. However, if we want to allow leakage of partial information, we need to modify the notion of equivalence.

Before introducing our notion of equivalence of expressions, we postulate an equivalence notion $\equiv_\mathbf{K}$ on the set of keys, and another equivalence, $\equiv_\mathbf{C}$ on the set of *valid* encryption terms. The word *valid*, defined precisely below, is meant for those encryption terms (and expressions) that "make sense". The equivalence on the set of valid expressions will be defined with the help of $\equiv_\mathbf{K}$ and $\equiv_\mathbf{C}$.

The reason for postulating equivalence on the set of keys is that we want to allow many key-generation processes in the probabilistic setting. We therefore have to be able to distinguish formal keys that were generated by different key-generation processes. We assume that an equivalence relation $\equiv_\mathbf{K}$ is given on the set of keys such that each equivalence class contains infinitely many keys. Let $\mathcal{Q}_{\mathbf{Keys}} := \mathbf{Keys} / \equiv_\mathbf{K}$.

Definition 31 (Key-Renaming Function). *A bijection $\sigma : \mathbf{Keys} \to \mathbf{Keys}$ is called* key-renaming function *if $\sigma(K) \equiv_\mathbf{K} K$ for all $K \in \mathbf{Keys}$. For any expression M, $M\sigma$ denotes the expression obtained from M by replacing all occurrences of keys K in M by $\sigma(K)$.*

The set **Exp** is often too big to suit our purposes. For example, sometimes we require that certain messages can be encrypted with certain keys only. We therefore define that a set of valid expressions satisfies at least the following properties:

Definition 32 (Valid Expressions). *A set of valid expressions is a subset* $\mathbf{Exp}_{\mathcal{V}}$ *of* $\mathbf{Exp}$ *such that:*

(i) all keys and all blocks are contained in $\mathbf{Exp}_{\mathcal{V}}$*;*
(ii) if $M \in \mathbf{Exp}_{\mathcal{V}}$*, then* $sub(M) \subset \mathbf{Exp}_{\mathcal{V}}$ *and all possible pairs of elements in* $sub(M)$ *are also in* $\mathbf{Exp}_{\mathcal{V}}$*;*
(iii) for any key-renaming function σ*,* $M \in \mathbf{Exp}_{\mathcal{V}}$ *iff* $M\sigma \in \mathbf{Exp}_{\mathcal{V}}$*.*

Given a set of valid expressions, the set of valid encryption terms is $\mathbf{Enc}_{\mathcal{V}} := \mathbf{Enc} \cap \mathbf{Exp}_{\mathcal{V}}$*.*

Example 33 (Valid Expressions for One-Time Pad). We introduce valid expressions that are suitable for the formal modeling of the One-Time Pad implementation discussed in Example 30. We assume that some length function $l : \mathbf{Keys} \rightarrow \{4, 5, \dots\}$ is given on the keys symbols. The length of a block is defined as $l(B) := |B| + 3$. We add 3 to match the length of the tag from Example 30. We define the length function on any expression in **Exp** by induction:

- $l((M, N)) := l(M) + 2l(N) + 1$,
- $l(\{M\}_K) := l(M) + 3$, if $l(M) = l(K) - 3$, and
- $l(\{M\}_K) := 0$, if $l(M) \neq l(K) - 3$.

The *valid expressions* are defined as those expressions in which the length of the encrypted subexpressions match the length of the encrypting key, and, in which no key is used twice to encrypt. (This latter condition is necessary to prevent leaking information because of the properties of the OTP.) Two keys are said to be equivalent according to $\equiv_{\mathbf{K}}$ iff l assigns the same length to them. Thus, we define the set of *valid expressions for OTP* as $\mathbf{Exp}_{\mathrm{OTP}} = \{M \in \mathbf{Exp} \mid M' \sqsubseteq M$ implies $l(M') > 0$, and each key encrypts at most once in $M\}$.

Equivalence of valid expressions is meant to incorporate the notion of security into the model: we want two expressions to be equivalent when they look the same to an adversary. If we think that the encryption is so secure that no partial information is revealed, then all undecryptable terms should look the same to an adversary. If partial information, say repetition of the encrypting key, or length is revealed, then we have to adjust the notion of equivalence accordingly. We do this by introducing an equivalence relation on the set of valid encryption terms in order to capture which ciphertexts an adversary cannot distinguish; in other words, what partial information (length, key, etc...) can an adversary retrieve from the ciphertext.

Definition 34 (Equivalence of Encryption Terms). *We assume defined an equivalence relation,* $\equiv_{\mathbf{C}}$ *on the set of valid encryption terms, with the property that for any* $M, N \in \mathbf{Enc}_{\mathcal{V}}$ *and key-renaming function* σ*,* $M \equiv_{\mathbf{C}} N$ *if and only if* $M\sigma \equiv_{\mathbf{C}} N\sigma$*.*

Let $\mathcal{Q}_{\mathbf{Enc}} := \mathbf{Enc}_{\mathcal{V}} \big/ \equiv_{\mathbf{C}}$*.*

Since we required that $M \equiv_{\mathbf{C}} N \in \mathbf{Enc}_{\mathcal{V}}$ if and only if $M\sigma \equiv_{\mathbf{C}} N\sigma$ whenever σ is a key-renaming function, σ induces a renaming on $\mathcal{Q}_{\mathbf{Enc}}$, which we also denote by σ.

Example 35 (Length-Revealing). Using the length-function of Definition 19, we can consider encrypted terms to be indistinguishable for an adversary if and only if the encrypted messages have the same length. For this case, we define $\equiv_{\mathbf{C}}$ so that it equates encryption terms with the same length, that is, $\{M\}_K \equiv_1 \{M'\}_{K'}$ if and only if $\ell(M) = \ell(M')$ (where we used $\equiv_1$ to denote this particular equivalence). An element of $\mathcal{Q}_{\mathbf{Enc}} = \mathbf{Enc}_{\mathcal{V}} \big/ \equiv_1$ contains all encryption terms for which the encrypted message has a specific length.

Example 36 (Which-Key Revealing). We can also consider the situation when an adversary can recognize that two encryption terms were encrypted with different keys. For this case, we need to define $\equiv_{\mathbf{C}}$ (which we denote now with $\equiv_2$) so that two encryption terms are equivalent if and only if they are encrypted with the same key, that is, $\{M\}_K \equiv_2 \{M'\}_{K'}$ if and only if $K = K'$.

Example 37 (One-Time Pad). As in the previous cases, we must find a suitable equivalence relation for formal expressions. One possibility is to label boxes again with the encrypting keys. Another possibility is to label the boxes with the length as well. In the OTP scheme, the key reveals the length of the ciphertext. Therefore, we can use the first, that is a simpler possibility. For this case, we define $\equiv_{\mathbf{C}}$ (which we denote now with $\equiv_{OTP}$) so that two encryption terms are equivalent if and only if they are encrypted with the same key, that is, $\{M\}_K \equiv_{OTP} \{M'\}_{K'}$ if and only if $K = K'$. This is almost the same as the which-key revealing case, except that the set of valid expressions is different.

Definition 38 (Formal Logic of Symmetric Encryption). *A formal logic for symmetric encryption is a triple $\Delta = (\mathbf{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ where $\mathbf{Exp}_{\mathcal{V}}$ is a set of valid expressions, $\equiv_{\mathbf{K}}$ is an equivalence relation on **Keys**, and $\equiv_{\mathbf{C}}$ is an equivalence relation on $\mathbf{Enc}_{\mathcal{V}}$. We require the elements of $\mathcal{Q}_{\mathbf{Keys}}$ to be infinite sets, and that for any key renaming function σ relative to $\mathcal{Q}_{\mathbf{Keys}}$,*

(i) if $M \in \mathbf{Exp}$, then $M \in \mathbf{Exp}_{\mathcal{V}}$ if and only if $M\sigma \in \mathbf{Exp}_{\mathcal{V}}$;
(ii) if $M, N \in \mathbf{Enc}_{\mathcal{V}}$, then $M \equiv_{\mathbf{C}} N$ if and only if $M\sigma \equiv_{\mathbf{C}} N\sigma$;
(iii) replacing an encryption term within a valid expression with another equivalent valid encryption term results in a valid expression.

To define the equivalence of expressions, we first assign to each valid expression an element in the set of *patterns*, **Pat**, defined the following way:

Definition 39 (Pattern). *We define the set of* patterns*, **Pat**, by the grammar:*

$$\mathbf{Pat} ::= \mathbf{Keys} \mid \mathbf{Blocks} \mid (\mathbf{Pat}, \mathbf{Pat}) \mid \{\mathbf{Pat}\}_{\mathbf{Keys}} \mid \Box_{\mathcal{Q}_{\mathbf{Enc}}}$$

The pattern of a valid expression M, denoted by pattern(M), is derived from M by replacing each undecryptable term $\{M'\}_K \sqsubseteq M$ ($K \notin$ R-Keys(M)) by $\Box_{\mu(\{M'\}_K)}$, where $\mu(\{M'\}_K) \in \mathcal{Q}_{\mathbf{Enc}}$ denotes the equivalence class containing $\{M'\}_K$.

Definition 40 (Equivalence of Expressions). *We say that two valid expressions M and N are* equivalent*, denoted by $M \cong N$, if there exists a key-renaming function σ such that pattern$(M) =$ pattern$(N\sigma)$. For a pattern Q, $Q\sigma$ denotes the pattern obtained by renaming all the keys and the box-indexes (which are equivalence classes in $\mathcal{Q}_{\mathbf{Enc}}$) in Q with σ.*

Example 41 (Which-Key Revealing and One-Time Pad). In the case when the elements of $\mathcal{Q}_{\mathbf{Enc}}$ contain encryption terms encrypted with the same key, Example 36, there is a one-to-one correspondence between $\mathcal{Q}_{\mathbf{Enc}}$ and **Keys**, and therefore we can index the boxes with keys instead of the elements in $\mathcal{Q}_{\mathbf{Enc}}$: $\Box_K$, $K \in \mathbf{Keys}$. If

$$N = ((\{0\}_{K_8}, \{K_2\}_{K_1}), ((K_7, \{(\{101\}_{K_9}, \{K_8\}_{K_5})\}_{K_5}), \{K_5\}_{K_7})),$$

the pattern according to the above definition is

$$\mathit{pattern}_2(N) = ((\{0\}_{K_8}, \Box_{K_1}), ((K_7, \{(\Box_{K_9}, \{K_8\}_{K_5})\}_{K_5}), \{K_5\}_{K_7})).$$

Then, two expressions are equivalent, if their patterns given by *pattern*$_2$ are the same up to key renaming. Let $\cong_2$ denote this equivalence on **Exp**. The pattern for OTP is the same as *pattern*$_2$, except that it is defined on the set of valid expression from Example 33. We denote it by *pattern*$_{\mathrm{OTP}}$, and the resulting equivalence of valid expressions by $\cong_{\mathrm{OTP}}$.

Example 42 (Length Revealing). For the case of length revealing, boxes can be indexed by the formal length, and two boxes are identical if their index is the same. That is, the pattern of N in the previous example is

$$\mathit{pattern}_1(N) = ((\{0\}_{K_8}, \Box_{\ell(K_2)}), ((K_7, \{(\Box_{\ell(101)}, \{K_8\}_{K_5})\}_{K_5}), \{K_5\}_{K_7})),$$

and two expressions are equivalent if and only if their patterns outside the boxes are the same, up to key renaming, and the boxes in the corresponding places are equal according to the lengths. Let $\cong_1$ denote this definition of equivalence on **Exp**.

Proper Equivalence of Ciphers. In order to be able to prove soundness and completeness we need to have some restrictions on $\equiv_{\mathbf{C}}$. The condition that we found the most natural for our purposes is called *proper equivalence* and is defined next. This condition is enough for both soundness and completeness.

Definition 43 (Proper Equivalence of Ciphers). *We say that an equivalence relation $\equiv_{\mathbf{C}}$ on $\mathbf{Enc}_{\mathcal{V}}$ is proper, if for any finite set of keys S, if $\mu \in \mathcal{Q}_{\mathbf{Enc}}$ contains an encryption term $\{N\}_K$ with $K \notin S$, then μ also contains an element C such that Keys$(C) \cap S = \emptyset$, and $K \not\sqsubseteq C$.*

In other words, for any finite set of keys S, if the equivalence class μ contains an element $\{N\}_K$ for some K not in S, then it is possible to find in μ another representative C such that no keys of C are in S, and K appears in C at most as an encrypting key. This way, fixing a set of keys S, one can obtain representatives for all classes that do not contain keys from S. In particular this implies that if μ has infinitely many encrypting keys, that is, the set

$$\mu_{\text{key}} := \{K \in \textbf{Keys} \mid \text{there is a valid expression } M \text{ such that } \{M\}_K \in \mu\}$$

is infinite, then there is an element in μ, in which no keys from $S \cup \{K\}$ appear. In fact, we show in Proposition 46 that the cardinality of the set μ_{key} is either 1 or ∞.

Example 44 (Which-Key Revealing). Relation $\equiv_2$ of Example 36 (*i.e.*two ciphers are equivalent iff they have the same encrypting key) is clearly proper. If $\{M\}_K \in \mu$, $K \notin S$, then $C = \{K'\}_K$ works for any $K' \notin S$; there is such a K', since we assumed that there are infinitely many keys. Choosing $C = \{B\}_K$ ($B \in$ **Blocks**) also works since **Blocks** is not empty. The same is true for OTP, but we have to require $\ell(K') = \ell(K) - 3$.

Example 45 (Length Revealing). Relation $\equiv_1$ of Example 35 (*i.e.*two ciphers are equivalent if and only if the encrypted messages have the same length) is clearly proper. If $\{M\}_K \in \mu$, $K \notin S$, then a good choice is $C = \{M'\}_K$ where M' is constructed by assigning to each key in M different from K, a new key K' not in S. We can do this since we assumed that there are infinitely many keys. Then, since key renaming does not change the length, $\ell(M) = \ell(M')$, and hence properness follows.

The following propositions will be useful for proving our general soundness and completeness results.

Proposition 46. *Let $\Delta = (\textbf{\textit{Exp}}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be such that $\equiv_{\mathbf{C}}$ is proper. The equivalence relation $\equiv_{\mathbf{C}}$ is such that for any equivalence class $\mu \in \mathcal{Q}_{\textbf{\textit{Enc}}}$, μ_{key} has either one, or infinitely many elements. Moreover, if σ is a key renaming that does nothing else but switches two keys L_1 and L_2, then $\{M\}_L\sigma \equiv_{\mathbf{C}} \{M\}_L$ as long as $L \neq L_1, L_2$ when $|\mu(\{M\}_L)_{\text{key}}| = 1$.*

Proof. Let $\mu \in \mathcal{Q}_{\textbf{Enc}}$, and assume that there are more than one encrypting keys in μ_{key}, that is, there are at least two different keys L and L_1 such that $\{M\}_L, \{M_1\}_{L_1} \in \mu$ for some valid expressions M and M_1. Since $\equiv_{\mathbf{C}}$ is proper and $\{M_1\}_{L_1} \in \mu$, if we consider $S = \{L\}$ ($L_1 \neq L$ thus $L_1 \notin S$) then μ has an element of the form $\{M'\}_{L'}$ in which no key of S appears and in which L_1 may only appear as an encrypting key. Therefore, we have that

$$L \notin \textit{Keys}(\{M'\}_{L'}). \tag{1}$$

Since we assumed that each equivalence class in $\mathcal{Q}_{\textbf{Keys}}$ contains infinitely many elements (recall Definition 38), there is a key $K \neq L$ such that $K \equiv_{\mathbf{K}} L$, and

$$K \notin \textit{Keys}(\{M\}_L) \cup \textit{Keys}(\{M'\}_{L'}). \tag{2}$$

Defining σ to do nothing else but to switch the keys K and L, we have that $\{M\}_L\sigma = \{M\sigma\}_K$ and (by (1) and (2)) $\{M'\}_{L'}\sigma = \{M'\}_{L'}$. By construction, we also have that $\{M\}_L \equiv_{\mathbf{C}} \{M'\}_{L'}$ which implies by Definition 34 that $\{M\}_L\sigma \equiv_{\mathbf{C}} \{M'\}_{L'}\sigma$. Merging these three results we obtain by transitivity $\{M\sigma\}_K \equiv_{\mathbf{C}} \{M'\}_{L'}$. Since

$\{M'\}_{L'} \in \mu$, it must hold that $\{M\sigma\}_K \in \mu$. Therefore, there are infinitely many encrypting keys in μ since there are infinitely many choices for K.

For the second part of the proof, suppose that σ is a key-renaming function that only switches the values of L_1 and L_2. If $\{M\}_L$ is an encryption term such that $|\mu(\{M\}_L)_{\mathrm{key}}| = 1$, and L_1, L_2 are such that $L \neq L_1, L_2$, then using properness and $S = \{L_1, L_2\}$, there is an encryption term $C \equiv_{\mathbf{C}} \{M\}_L$ such that $\mathit{Keys}(C) \cap \{L_1, L_2\} = \emptyset$. Then $C\sigma = C$, and $\{M\}_L\sigma \equiv_{\mathbf{C}} C\sigma = C \equiv_{\mathbf{C}} \{M\}_L$.

For the other case, when $|\mu(\{M\}_L)_{\mathrm{key}}| = \infty$, consider a new term $\{M'\}_{L'} \equiv_{\mathbf{C}} \{M\}_L$ such that $L' \notin \{L, L_1, L_2\}$. Applying the same reasoning as above we obtain $\{M'\}_{L'}\sigma \equiv_{\mathbf{C}} \{M'\}_{L'}$, and $\{M\}_L\sigma \equiv_{\mathbf{C}} \{M'\}_{L'}\sigma \equiv_{\mathbf{C}} \{M'\}_{L'} \equiv_{\mathbf{C}} \{M\}_L$. $\square$

Proposition 47. *Let $\Delta = (\boldsymbol{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be such that $\equiv_{\mathbf{C}}$ is proper. If σ is a key-renaming function (relative to $\equiv_{\mathbf{K}}$), then for any $\mu \in \mathcal{Q}_{\boldsymbol{Enc}}$, $|\mu_{\mathrm{key}}| = |\sigma(\mu)_{\mathrm{key}}|$.*

Proof. If $|\mu_{\mathrm{key}}| = \infty$, then $|\sigma(\mu)_{\mathrm{key}}| = \infty$, since for any $\{M\}_K \in \mu$, $\{M\}_K\sigma = \{M\sigma\}_{\sigma(K)} \in \sigma(\mu)$. Since σ is a bijection, and since any μ contains either only one or infinitely many elements, the claim follows. $\square$

The next proposition states that if an equivalence relation $\equiv_{\mathbf{C}}$ is proper, then given a set of valid ciphers $\{\{N_i\}_{L_i}\}_{i=1}^n$, corresponding to equivalence classes $\mu_1, \ldots, \mu_n$ (eventually repeated), such that none of the L_is are in S, then it is possible to choose a representative of each μ_j, denoted by C_j, such that no key of S occurs in any C_j, the L_is occur at the most as encrypting keys in the C_js, and no key occurs in two C_js unless the corresponding two equivalence classes both have the same, single, encrypting key. Intuitively, one can construct a representative for each class that is independent of the representatives of all the other classes (except the case when there is a single encrypting key in the class), and at the same time independent of S.

Proposition 48. *Let $\Delta = (\boldsymbol{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be such that $\equiv_{\mathbf{C}}$ is proper, and let $\mathfrak{C} = \{\{N_i\}_{L_i}\}_{i=1}^n$ be a set of valid ciphers. Let $\mu_1, \ldots, \mu_n$ denote the equivalence-classes of all elements in $\mathfrak{C}$ with respect to $\equiv_{\mathbf{C}}$.*

If S is a finite set of keys such that $L_i \notin S$ for all $i \leq n$, then for each μ_j there is an element $C_j \in \mu_j$ such that

(i) *$\mathit{Keys}(C_j) \cap S = \emptyset$,*
(ii) *$L_i \not\sqsubseteq C_j$ for all $i \in \{1, \ldots, n\}$, and*
(iii) *if $\mu_j = \mu_i$, then $C_j = C_i$; if $\mu_j \neq \mu_i$, then $\mathit{Keys}(C_j) \cap \mathit{Keys}(C_i) \neq \emptyset$ if and only if $(\mu_j)_{\mathrm{key}} = (\mu_i)_{\mathrm{key}} = \{K\}$ for some key K. In this case $\mathit{Keys}(C_j) \cap \mathit{Keys}(C_i) = \{K\}$, $K \not\sqsubseteq C_j$, and $K \not\sqsubseteq C_i$.*

Proof. Proof goes by induction. The statement is clearly true if $n = 1$, since $\equiv_{\mathbf{C}}$ is proper. Suppose it is true for $n-1$. Let $\{N_1\}_{L_1}$, $\{N_2\}_{L_2}, \ldots, \{N_n\}_{L_n}$ be valid expressions, and let S be a set of keys such that $L_i \notin S$. Without loss of generality, we can assume, that the numbering is such that there is an l, $1 \leq l \leq n$, such that

$$|(\mu_j)_{\mathrm{key}}| = \begin{cases} 1 & \text{if } j \leq l \\ \infty & \text{if } j > l. \end{cases}$$

Case 1: Let us first assume that $l = n$ and that there is an $m \in \{1, \dots, n-1\}$ such that $L_n = L_m$. Since the statement is assumed to be true for $n-1$, we can choose C_j for $j \leq n-1$ such that conditions (i), (ii), (iii) hold for these $\{C_j\}_{j=1}^{n-1}$ and S. If $\mu_n = \mu_j$ for some $j \leq n-1$, then there is nothing to prove, $C_n = C_j$ has already been chosen. If there is no such j, then consider

$$S_{n-1} := \left(\left(\bigcup_{j=1}^{n-1} \mathit{Keys}(C_j) \cup \bigcup_{i=1}^{n-1} \{L_i\} \right) \setminus \{L_n\} \right) \cup S$$

Given S_{n-1} and $\{N_n\}_{L_n}$, according to the assumption of properness of $\equiv_{\mathbf{C}}$, there is a $C \in \mu_n$ such that $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ and $L_n \not\sqsubseteq C$. Let us define $C_n := C$. Condition (i) follows from the fact that $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ and $S \subseteq S_{n-1}$; (ii) is true, since for all $j \leq n-1$ (a) $L_m \not\sqsubseteq C_j$ by the induction hypothesis (IH); (b) $L_n \not\sqsubseteq C_j$ because we assumed that $L_m = L_n$; and (c) $L_i \not\sqsubseteq C_j$ for $i \leq n-1$ also by IH. For $j = n$ we have $C_j = C_n = C$ hence (d) $L_m = L_n \not\sqsubseteq C$ by construction of C; and (e) $L_i \not\sqsubseteq C$ for $i \leq n-1$ and $i \neq m$, because either $L_i = L_n \not\sqsubseteq C$ by the previous case or $L_i \neq L_n$ and in this case, by construction of C, $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ and $L_i \in S_{n-1}$. Finally, for case (iii) we have by construction of C that $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ hence for $j \leq n-1$, $K \in \mathit{Keys}(C_j) \cap \mathit{Keys}(C)$ implies that $K = L_n$; but we have just proved in (ii) that $L_n \not\sqsubseteq C_j$ and $L_n \not\sqsubseteq C$ hence both C_j and C are of the form $\{\cdot\}_{L_n}$. Finally, since $l = n$, we have that $|(\mu_j)_{\text{key}}| = |(\mu_n)_{\text{key}}| = 1$, hence $(\mu_j)_{\text{key}} = (\mu_n)_{\text{key}} = \{L_n\}$. The converse is immediate. The case $i, j \leq n-1$ follow immediately by IH.

Case 2: Suppose now that $l = n$ but there is no $m \in \{1, \dots, n-1\}$ such that $L_n = L_m$. We have that $\{\{N_i\}_{L_i}\}_{i=1}^{n-1}$ and $S' := S \cup \{L_n\}$ satisfy the conditions of the IH. We can then choose C_j for $j \leq n-1$ such that conditions (i), (ii), (iii) hold replacing S by S'. Again, if $\mu_n = \mu_j$ for some $j \leq n-1$, then $C_n = C_j$ has already been chosen. Condition (i) follows because $\mathit{Keys}(C_j) \cap S \subseteq \mathit{Keys}(C_j) \cap S' = \emptyset$ by IH; (ii) holds because for all $j \leq n-1$ (a) $L_n \not\sqsubseteq C_j$ because by IH we have $\mathit{Keys}(C_j) \cap (S \cup \{L_n\}) = \emptyset$; and (b) $L_i \not\sqsubseteq C_j$ for $i \leq n-1$ by IH. For $j = n$ we have $C_n = C_k$ for some $k \leq n-1$ hence (c) $L_n \not\sqsubseteq C_n = C_k$ because of case (b); and (d) $L_i \not\sqsubseteq C_n = C_k$ for $i \leq n-1$ also by IH. Finally (iii) follows from IH because $\mu_n = \mu_j$ for some $j \leq n-1$.

If there is no such j, then consider

$$S_{n-1} := \left(\bigcup_{j=1}^{n-1} \mathit{Keys}(C_j) \cup \bigcup_{i=1}^{n-1} \{L_i\} \right) \cup S$$

We should first notice that $L_n \notin S_{n-1}$: by IH, $\mathit{Keys}(C_j) \cap S' = \mathit{Keys}(C_j) \cap (S \cup \{L_n\}) = \emptyset$ for $j \leq n-1$; by hypothesis $L_n \neq L_m$ for $m \leq n-1$, and by hypothesis of the proposition $L_n \notin S$.

Given S_{n-1} and $\{N_n\}_{L_n}$, according to the assumption of properness of $\equiv_{\mathbf{C}}$, there is a $C \in \mu_n$ such that $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ and $L_n \not\sqsubseteq C$. Let us define $C_n := C$. Condition (i) follows from the fact that $\mathit{Keys}(C) \cap S_{n-1} = \emptyset$ and $S \subseteq S_{n-1}$; (ii) is true, since for all $j \leq n-1$ (a) $L_n \not\sqsubseteq C_j$ because by IH, $\mathit{Keys}(C_j) \cap (S \cup \{L_n\}) = \emptyset$; and (b) $L_i \not\sqsubseteq C_j$ for $i \leq n-1$ also by IH. For $j = n$ we have $C_n = C$ hence (c)

$L_n \not\sqsubseteq C$ by construction of C; and (d) $L_i \not\sqsubseteq C$ for $i \leq n-1$ because $L_i \in S_{n-1}$ and by construction of C, *Keys*$(C) \cap S_{n-1} = \emptyset$. Finally, (iii) follows as well, because by construction of C, *Keys*$(C) \cap S_{n-1} = \emptyset$ implies *Keys*$(C) \cap$ *Keys*$(C_j) = \emptyset$ for all $j \leq n-1$. The cases $i, j \leq n-1$ follow immediately by IH.

Case 3: Suppose now that $l < n$ and that there is an $m \in \{1, \ldots, n-1\}$ such that $L_n = L_m$. Select $\{N'_n\}_{L'_n}$ such that $\{N'_n\}_{L'_n} \equiv_{\mathbf{C}} \{N_n\}_{L_n}$ and $L'_n \notin \{L_i\}_{i=i}^{n}$. This is possible since in this case we have that $|(\mu_n)_{\mathrm{key}}| = \infty$.

One now has that $\{\{N_i\}_{L_i}\}_{i=1}^{n-1} \cup \{N'_n\}_{L'_n}$ and $S' := S \cup \{L'_n\}$ satisfy the conditions of the IH and the result follows similarly to the previous case.

Case 4: Suppose now that $l < n$ but there is no $m \in \{1, \ldots, n-1\}$ such that $L_n = L_m$. Again, $\{\{N_i\}_{L_i}\}_{i=1}^{n-1}$ and $S' := S \cup \{L_n\}$ satisfy the conditions of the IH and the result follows similarly to Case 2. ☐

Given sets $\mathfrak{C}$ and S as in the conditions of the proposition, let $\mu(\mathfrak{C})$ denote the set of equivalence classes of the elements in $\mathfrak{C}$ and let $\mathfrak{R}(\mathfrak{C}, S)$ denote the nonempty set

$$\mathfrak{R}(\mathfrak{C}, S) := \left\{ \{C_\nu\}_{\nu \in \mu(\mathfrak{C})} \;\middle|\; \begin{array}{l} C_\nu \in \nu, \text{ and } \{C_\nu\}_{\nu \in \mu(\mathfrak{C})} \text{ and } S \text{ satisfy conditions} \\ \text{(i), (ii), and (iii) of Proposition 48} \end{array} \right\}$$

We will need the following proposition for the general completeness theorem. For $\mu \in \mathcal{Q}_{\mathbf{Enc}}$, let $\|\mu_{\mathrm{key}}\| = \mu_{\mathrm{key}}$ if $|\mu_{\mathrm{key}}| = 1$, and $\|\mu_{\mathrm{key}}\| = \emptyset$ if $|\mu_{\mathrm{key}}| = \infty$ (by Proposition 46, the only possible cases).

Proposition 49. *Let $\Delta = (\boldsymbol{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be such that $\equiv_{\mathbf{C}}$ is proper, S, U finite sets of keys, and $C = \{M\}_L$ an encryption term such that Keys$(C) \cap S = \emptyset$. Then, for any key-renaming function σ, there is a key-renaming function σ' such that:*

(i) σ' is the identity map on $S \setminus \|\mu(C\sigma)_{\mathrm{key}}\|$;
(ii) $\sigma'\big(\mathit{Keys}(C) \setminus \|\mu(C)_{\mathrm{key}}\|\big) \cap (S \cup U) = \emptyset$;
(iii) $C\sigma \equiv_{\mathbf{C}} C\sigma'$; and
(iv) σ' changes only finitely many keys.

Proof. Let $S' = S \setminus \|\mu(C\sigma)_{\mathrm{key}}\|$ and S'' be a set of keys such that each equivalence class of $\equiv_{\mathbf{K}}$ has the same number of elements in S' and S'' (possible because we required each class to have an infinite number of keys) and $S'' \cap (S' \cup$ *Keys*$(C) \cup$ *Keys*$(C\sigma)) = \emptyset$. Let σ_1 be a key-renaming function that switches the keys of S' and S'' and leaves all other keys unchanged. Let σ_2 be defined by cases: if $\|\mu(C\sigma)_{\mathrm{key}}\| = \{\sigma(L)\}$, let σ_2 be the same as σ on *Keys*(C), map elements of *Keys*$(C\sigma) \setminus$ *Keys*(C) bijectively to *Keys*$(C) \setminus$ *Keys*$(C\sigma)$, and be the identity map elsewhere; if $\|\mu(C\sigma)_{\mathrm{key}}\| = \emptyset$, let σ_2 be the same as above except that $\sigma_2(L)$ is defined to be any key $K' \notin S' \cup S'' \cup$ *Keys*$(C) \cup$ *Keys*$(C\sigma)$, $\sigma_2(K') = L$ and $\sigma_2(\sigma(L)) = \sigma(L)$ (these last two for σ_2 to remain a bijection). In either case σ_2 is the identity map on S''.

Let $\sigma'' := \sigma_1^{-1} \circ \sigma_2 \circ \sigma_1$. Then, σ_1 first maps all elements of S' onto S'', which are then unchanged by σ_2, and finally they are mapped back to S' by σ_1^{-1}. Hence Condition (i) holds for σ''.

Let us now prove that σ'' satisfies Condition (iii), that is, $C\sigma \equiv_{\mathbf{C}} C\sigma''$.

1. Since *Keys*(C) is disjoint from both S' and S'' we have that σ_1 is the identity map on *Keys*(C), therefore $\sigma_2(\sigma_1(K)) = \sigma_2(K)$ for all $K \in$ *Keys*(C).
   (a) If $\|\mu(C\sigma)_{\mathrm{key}}\| \neq \emptyset$, then $\sigma_2(K) = \sigma(K)$ for all $K \in$ *Keys*(C), hence $\sigma_2(\sigma_1(K)) = \sigma_2(K) = \sigma(K)$ for all $K \in$ *Keys*(C). Hence $C\sigma_1\sigma_2 = C\sigma$.
   (b) If $\|\mu(C\sigma)_{\mathrm{key}}\| = \emptyset$, then $\sigma_2(K) = \sigma(K)$ for all $K \in$ *Keys*$(C) \setminus \{L\}$. Let us define ρ that switches keys K' and $\sigma(L)$, and is the identity elsewhere. By Proposition 46, $C\sigma_1\sigma_2\rho \equiv_{\mathbf{C}} C\sigma_1\sigma_2$. Let $K \in$ *Keys*(C). If $K = L$, then $\rho(\sigma_2(\sigma_1(L))) = \rho(\sigma_2(L)) = \rho(K') = \sigma(L)$. If $K \neq L$, then $\rho(\sigma_2(\sigma_1(K))) = \rho(\sigma_2(K)) = \rho(\sigma(K)) = \sigma(K)$ since $\sigma(K) \notin \{K', \sigma(L)\}$. Hence $C\sigma_1\sigma_2\rho = C\sigma$ and by transitivity we obtain $C\sigma \equiv_{\mathbf{C}} C\sigma_1\sigma_2$.

   In both cases we obtain $C\sigma \equiv_{\mathbf{C}} C\sigma_1\sigma_2 (= \{M\sigma_2\}_{\sigma_2(L)})$.
2. By properness of $\equiv_{\mathbf{C}}$ and $\sigma_2(L) \notin S' \cup S''$, there is an encryption term $C' \equiv_{\mathbf{C}} C\sigma_1\sigma_2$ such that *Keys*$(C') \cap (S' \cup S'') = \emptyset$. Therefore, $C' = C'\sigma_1^{-1}$.
3. Finally, by definition of $\equiv_{\mathbf{C}}$ and 2 we also have that $C'\sigma_1^{-1} \equiv_{\mathbf{C}} C\sigma_1\sigma_2\sigma_1^{-1}$ and so $C\sigma \equiv_{\mathbf{C}} C\sigma_1\sigma_2 \equiv_{\mathbf{C}} C' \equiv_{\mathbf{C}} C'\sigma_1^{-1} \equiv_{\mathbf{C}} C\sigma_1\sigma_2\sigma_1^{-1} = C\sigma''$ by definition of σ''.

Condition (iv) holds trivially from the way we constructed σ''.

Finally, we construct σ' from σ'' such that $\sigma'\big(\textit{Keys}(C) \setminus \|\mu(C)_{\mathrm{key}}\|\big) \cap (S \cup U) = \emptyset$. We show that this construction carries the properties (i), (iii) and (iv) of σ'' over to σ'.

If for all $K \in$ *Keys*$(C) \setminus \|\mu(C)_{\mathrm{key}}\|$ we have that $\sigma''(K) \notin (S \cup U)$, then let $\sigma' = \sigma''$. Obviously this σ' satisfies Conditions (i)-(iv).

If there is a $K \in$ *Keys*$(C) \setminus \|\mu(C)_{\mathrm{key}}\|$ such that $\sigma''(K) = K'' \in (S \cup U)$, then pick any $L_2 \equiv_{\mathbf{K}} K$, such that $L_2 \notin (S \cup U \cup S'' \cup \textit{Keys}(C))$ and $\sigma''(L_2) = L_2$, that is, L_2 is a key that was never used before. It is possible to choose L_2 as all these sets are finite. Let ρ be a key-renaming function that switches L_2 and K and consider the substitution $\rho\sigma'' = \sigma'' \circ \rho$. This substitution is equal to σ'' except for the values given to L_2 and K. Let us prove Conditions (i)-(iv) for $\rho\sigma''$.

To prove (i) note that $L_2 \notin S \supseteq S'$, and $K \in$ *Keys*(C), but *Keys*$(C) \cap S = \emptyset$. For all other keys $\rho\sigma''$ is equal to σ'' that is the identity in S'. For (ii) we have by Proposition 46 that $C\rho \equiv_{\mathbf{C}} C$ hence $C\rho\sigma'' \equiv_{\mathbf{C}} C\sigma'' \equiv_{\mathbf{C}} C\sigma$. Note that if $|\mu(C)_{\mathrm{key}}| = 1$, then $\|\mu(C)_{\mathrm{key}}\| = \{L\}$, so $K \neq L$ by hypothesis and $L_2 \neq L \in$ *Keys*(C) by construction. It is obvious that $\rho\sigma''$ only changes finitely many keys as σ'' does. Finally, $(\rho\sigma'')\big(\textit{Keys}(C) \setminus \|\mu(C)_{\mathrm{key}}\|\big) \cap (S \cup U) \subset \sigma''\big(\textit{Keys}(C) \setminus \|\mu(C)_{\mathrm{key}}\|\big) \cap (S \cup U)$ as we removed one key from the intersection, $\sigma''(\rho(K)) = L_2 \notin (S \cup U)$.

Define then the new σ'' as $\rho\sigma''$ and iterate this procedure until all coincidences with $(S \cup U)$ are removed. At each step one key is removed from the intersection. This procedure terminates as *Keys*(C) is finite and then the resulting renaming function is the desired σ' as it satisfies Conditions (i)-(iv). $\square$

One may ask if we can make σ' identity in all S. However, that is not possible when $\sigma(L) \in S$ and $|\mu(C\sigma)_{\mathrm{key}}| = 1$. As an example, consider the relation $\equiv_{\mathbf{C}}$ of Example 36 (two terms are equivalent if and only if they have the same encrypting key), $S = \{L_1\}$, and $C = \{M\}_L$. Given the substitution σ that switches L and L_1 one will never be able to create σ' such that $C\sigma = \{M\sigma\}_{L_1} \equiv_{\mathbf{C}} C\sigma'$ because one will always need $\sigma'(L) = L_1$ to obtain the equivalence. This problem does not occur when $|\mu(C\sigma)_{\mathrm{key}}| = \infty$ because we have an infinite number of keys to map the encrypting key L.

5.3 Interpretation

To each valid formal expression M, the interpretation assigns a random variable $\Phi(M)$ taking values in $\overline{\mathbf{strings}}$. We do not give one specific interpretation function though, we will just say that a function Φ is an interpretation if it satisfies certain properties. We assume, that a function ϕ is fixed in advance, which assigns to each formal key a key-generation algorithm. If $\Phi(B) \in \overline{\mathbf{strings}}$ (constant random variable) is given for blocks, then, the rest of Φ is determined the following way: First, run the key-generation algorithm assigned by ϕ for each key in *Keys*(M). Then, using the outputs of these key-generations, translate the formal expressions according to the following rules: For each key, use the output of the corresponding key-generation. For blocks, just use $\Phi(B)$. For each pair, apply $[\cdot,\cdot]$ to the interpretations of the expressions inside the formal pair. For each formal encryption, run the encryption algorithm using the key string that was output by the key generation, on the interpretation of the formal expression inside the formal encryption. The randomness of $\Phi(M)$ comes from the initial key-generation, and from running the encryption algorithm independently every time you encounter a formal encryption. The precise definition is quite technical and given in Definition 51, but it is probably clear enough from the following example:

Example 50. For $M = ((\{0\}_{K_{10}}, K_5), \{K_{10}\}_{K_5})$, the interpretation is $\Phi(M) : (\Omega_{\mathcal{E}} \times \Omega_{\mathcal{E}}) \times (\Omega_{\phi(K_5)} \times \Omega_{\phi(K_{10})}) \to \overline{\mathbf{strings}}$, where $\Phi(M)(\omega_1, \omega_2, \omega_3, \omega_4)$ is

$$\big[[\mathcal{E}(\phi(K_{10})(\omega_4), \Phi(0))(\omega_1), \phi(K_5)(\omega_3)], \mathcal{E}(\phi(K_5)(\omega_3), \phi(K_{10})(\omega_4))(\omega_2)\big].$$

There are four instances of randomness, two coming from the generation of keys by the key-generation algorithm (for K_5 and for K_{10}), and the other two from the encryptions $\{0\}_{K_{10}}$ and $\{K_{10}\}_{K_5}$.

Definition 51 (Interpretation of Formal Expressions). *Let $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$ be a general symmetric encryption scheme, with $\{(\Omega_{\mathcal{K}_i}, \mathrm{Pr}_{\mathcal{K}_i})\}_{i\in I}$ denoting the probability fields for key generation, and with $(\Omega_{\mathcal{E}}, \mathrm{Pr}_{\mathcal{E}})$ denoting the probability field for the randomness of encryption. Let **Exp**$_{\mathcal{V}}$ be a set of valid expressions. For each valid expression M, let the probability space $(\Omega_M, \mathrm{Pr}_M)$ be defined recursively as*

$$\begin{aligned}
&(\Omega_K, \mathrm{Pr}_K) := (\{\omega_0\}, \mathbf{1}_{\{\omega_0\}}) \text{ for } K \in \textbf{\textit{Keys}};\\
&(\Omega_B, \mathrm{Pr}_B) := (\{\omega_0\}, \mathbf{1}_{\{\omega_0\}}) \text{ for } B \in \textbf{\textit{Blocks}};\\
&(\Omega_{(M,N)}, \mathrm{Pr}_{(M,N)}) := (\Omega_M \times \Omega_N, \mathrm{Pr}_M \otimes \mathrm{Pr}_N);\\
&(\Omega_{\{M\}_K}, \mathrm{Pr}_{\{M\}_K}) := (\Omega_{\mathcal{E}} \times \Omega_M, \mathrm{Pr}_{\mathcal{E}} \otimes \mathrm{Pr}_M).
\end{aligned}$$

*Where $(\{\omega_0\}, \mathbf{1}_{\{\omega_0\}})$ is just the trivial probability-space with one elementary event, ω_0 only; the tensor product stands for the product probability. Suppose that a function ϕ : **Keys** $\to \{\mathcal{K}_i\}_{i\in I}$ is given assigning abstract keys to key generation algorithms, such that $\phi(K) = \phi(K')$ if and only if $K \equiv_{\mathbf{K}} K'$. Let $\iota : \{1, \ldots, |Keys(M)|\} \to Keys(M)$ be a bijection enumerating the keys in Keys(M). Let*

$$(\Omega_{Keys(M)}, \mathrm{Pr}_{Keys(M)}) := \\ \big(\, \Omega_{\phi(\iota(1))} \times \cdots \times \Omega_{\phi(\iota(|Keys(M)|))}, \mathrm{Pr}_{\phi(\iota(1))} \otimes \cdots \otimes \mathrm{Pr}_{\phi(\iota(|Keys(M)|))} \,\big).$$

The function $(M, M') \mapsto (\Phi_M(M') : \Omega_{M'} \times \Omega_{Keys(M)} \to \overline{\mathbf{strings}})$ *defined whenever* $M' \sqsubseteq M$*, is called* an interpreting function*, if it satisfies the following properties:*

> $\Phi_M(B)(\omega_0, \omega) = \Phi_N(B)(\omega_0, \omega')$ *for all* M*,* N *valid expressions,* $B \in$ ***Blocks****,* $B \sqsubseteq M$*,* $B \sqsubseteq N$*, and arbitrary* $\omega \in \Omega_{Keys(M)}$*,* $\omega' \in \Omega_{Keys(N)}$*. Let* $\Phi(B) := \Phi_M(B)$*.*
> $\Phi_M(K)(\omega_0, (\omega_1, \ldots, \omega_{|Keys(M)|})) = \phi(K)(\omega_{\iota^{-1}(K)})$ *for* $K \in Keys(M)$*, with* $\omega_j \in \Omega_{\phi(\iota(j))}$*.*
> $\Phi_M((M', M''))((\omega', \omega''), \omega) = [\Phi_M(M')(\omega', \omega), \Phi_M(M'')(\omega'', \omega)]$ *for all* $\omega' \in \Omega_{M'}$*,* $\omega'' \in \Omega_{M''}$*, and* $\omega \in \Omega_{Keys(M)}$ *if* $(M', M'') \sqsubseteq M$*.*
> $\Phi_M(\{M'\}_K)((\omega_{\mathcal{E}}, \omega'), \omega) = \mathcal{E}(\Phi_M(K)(\omega_0, \omega), \Phi_M(M')(\omega', \omega))(\omega_{\mathcal{E}})$ *for all* $\omega' \in \Omega_{M'}$*,* $\omega_{\mathcal{E}} \in \Omega_{\mathcal{E}}$*,* $\omega \in \Omega_{Keys(M)}$ *if* $\{M'\}_K \sqsubseteq M$*.*

Let $\Phi(M) := \Phi_M(M)$*, and let* $[\![M]\!]_\Phi$ *denote the distribution of* $\Phi(M)$*.*

Clearly, the definition is not necessarily well-defined depending on what $\mathrm{Dom}_{\mathcal{E}}$ is. We simply assume, that $\mathrm{Dom}_{\mathcal{E}}$ is such that this does not cause a problem, (another possibility is to restrict the set of valid expressions to those elements for which the interpretation is well-defined).

Example 52 (Interpretation for Computational Systems). We discussed the interpretation for computational systems in Section 3.4. The algorithm there includes boxes, which should be left out for now, and what remains is a special case of the general interpretation presented here, with the considerations of Example 29.

Example 53 (Interpretation for One-Time Pad). The interpretation of the valid expressions that we gave in Example 33 for the OTP is defined similarly to the computational case, with some minor changes regarding the tagging of the messages. Also, there is no security parameter in this encryption scheme, so the interpretation outputs a single random variable for each formal expression (rather than a family of such variables). We present here the full algorithm:

> **algorithm** INTERPRETATION$_{\mathrm{OTP}}(M)$
>   **for** $K \in Keys(M)$ **do** $\tau(K) \longleftarrow \mathcal{K}_{l(K)}$
> $y \longleftarrow \mathrm{CONVERT}_{\mathrm{OTP}}(M)$
>   **return** y
>
> **algorithm** CONVERT$_{\mathrm{OTP}}(N)$
>   **if** $N = K$ where $K \in$ **Keys then**
>   **return** $\tau(K)$
>   **if** $N = B$ where $B \in$ **Blocks then**
>   **return** $\langle B, 100\rangle$
>   **if** $N = (N_1, N_2)$ **then**
>   **return** $[\mathrm{CONVERT}_{\mathrm{OTP}}(N_1), \mathrm{CONVERT}_{\mathrm{OTP}}(N_2)]$
>   **if** $N = \{N_1\}_K$ **then**
>   **return** $\langle \mathcal{E}(\tau(K), \mathrm{CONVERT}_{\mathrm{OTP}}(N_1)), 110\rangle$

5.4 Soundness

An interpretation assigns a random variable $\Phi(M)$ (and the distribution $[\![M]\!]_\Phi$ of $\Phi(M)$) to a formal valid expression M. On the set of valid expressions the equivalence $\cong$ equates expressions that a formal adversary supposedly cannot distinguish, whereas the equivalence $\approx$ equates random variables (and distributions) that a probabilistic adversary is not supposed to be able to distinguish. The question is, how the formal and the probabilistic equivalence are related through the interpretation. We say that soundness holds if $M \cong N$ implies $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$, whereas we say that completeness holds if $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$ implies $M \cong N$.

The key to a soundness theorem is to have enough boxes in the definition of formal equivalence, *i.e.*, there should be enough elements in $\mathcal{Q}_{\mathbf{Enc}}$. It is clear that in the extreme case, when the equivalence on encryption terms, $\equiv_{\mathbf{C}}$, is defined so that two encryption terms are equivalent iff they are the same, soundness holds trivially for all interpretations; but this would be completely impractical, it assumes a formal adversary that can see everything inside every encryption. It is also immediate, that if soundness holds with a given $\equiv_{\mathbf{C}}$ (and a given interpretation), and $\equiv'_{\mathbf{C}}$ is such that for any two encryption terms M and N, $M \equiv'_{\mathbf{C}} N$ implies $M \equiv_{\mathbf{C}} N$ (*i.e.*$\equiv'_{\mathbf{C}}$ has more boxes), then, keeping the same interpretation, soundness holds with the new $\equiv'_{\mathbf{C}}$ as well. Hence, in a concrete situation, the aim is to introduce enough boxes to achieve soundness, but not too many, to sustain practicality. One way to avoid having too many boxes is to require at the same time completeness: we will see later, that obtaining completeness requires that we do not have too many boxes.

The following theorem claims the equivalence of two conditions. It is almost trivial that condition (i) implies condition (ii). The claim that (ii) implies (i) can be summarized the following way: if soundness holds for pairs of valid expressions M' and N' with a special relation between them (described in (ii)), then soundness holds for all expressions (provided that they do not have encryption cycles). In other words, if $M' \cong N'$ implies $[\![M']\!]_\Phi \approx [\![N']\!]_\Phi$ for certain specified pairs M' and N', then $M \cong N$ implies $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$ for any two pairs of valid expressions M and N. The relation between M' and N' is that N' is obtained from M' via substitution of all undecryptable terms (that are encrypted with some key K) by the representative of its equivalence class.

For definition of key cycles and *B-Keys*, see Section 3.1. $\mathfrak{R}(\mathfrak{C}, S)$, is defined in Section 5.2. Given an encryption term $\{N\}_K$, we denote by $\mu(\{N\}_K)$ its equivalence class and by $C_{\mu(\{N\}_K)}$ a representative of its class.

Theorem 54 (Soundness). *Let $\Delta = (\boldsymbol{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be a formal logic for symmetric encryption such that $\equiv_{\mathbf{C}}$ is proper and for each $M \in \boldsymbol{Exp}_{\mathcal{V}}$, B-Keys$(M)$ is not cyclic in M. Let $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$ be a general encryption scheme, and Φ an interpretation of $\boldsymbol{Exp}_{\mathcal{V}}$ in Π. The following conditions are equivalent:*

(i) Soundness holds for Φ: $M \cong N$, implies $\Phi(M) \approx \Phi(N)$.
(ii) For any set of valid encryption terms $\mathfrak{C} = \{\{N_i\}_{L_i}\}_{i=1}^n$, and finite set of keys S such that $L_i \notin S$ ($i \in \{1, \ldots, n\}$), there is an element $\{C_\nu\}_{\nu\in\mu(\mathfrak{C})}$ of $\mathfrak{R}(\mathfrak{C}, S)$ such that the followings hold:
if $\left\{\{N_{i_j}\}_K\right\}_{j=1}^{l} \subseteq \mathfrak{C}$ and $M \in \boldsymbol{Exp}_{\mathcal{V}}$ are such that
1. $\{N_{i_1}\}_K, \{N_{i_2}\}_K, \ldots, \{N_{i_l}\}_K \sqsubseteq M$,

2. K *does not occur anywhere else in* M,
3. *if* $\{M_1\}_L \in vis\,(M)$ *but* $L \notin$ *R-Keys*(M)*, then* $\{M_1\}_L \in \mathfrak{C} \cup \{C_\nu\}_{\nu \in \mu(\mathfrak{C})}$,
4. *R-Keys*$(M) \subseteq S$.

then, if we denote by M' *the expression obtained from* M *by replacing each* $\{N_{i_j}\}_K$ *with* $C_{\mu(\{N_{i_j}\}_K)}$*, we have that* $[\![M]\!]_\Phi \approx [\![M']\!]_\Phi$.

Proof. The proof of this theorem is motivated by the soundness proof in [3]. The idea of the proof is the following: starting from two acyclic expressions $M_0 = M \cong N = N_0$, we create expressions $M_1, \ldots, M_b$ and $N_1, \ldots, N_{b'}$ such that M_{i+1} is obtained from M_i via a replacement of encryption terms as described in condition (ii). Acyclicity ensures that the encrypting key of the replaced encryption terms will not occur anywhere else. Similarly for N_{i+1} and N_i. We do this so that M_b and $N_{b'}$ will differ only by key renaming. Then, by condition (ii), $[\![M_{i+1}]\!]_\Phi \approx [\![M_i]\!]_\Phi$, and $[\![N_{i+1}]\!]_\Phi \approx [\![N_i]\!]_\Phi$. But, $[\![M_b]\!]_\Phi = [\![N_{b'}]\!]_\Phi$, and therefore the theorem follows.

Now in detail. Condition (ii) follows from (i) easily: For any set $\{C_{\mu(\{N_{i_j}\}_K)}\}_{i=1}^{l}$ provided by Proposition 48, the encrypting key of $C_{\mu(\{N_{i_j}\}_K)}$ is not contained in S hence it is not a recoverable key of M. Therefore, while computing the pattern of M', $C_{\mu(\{N_{i_j}\}_K)}$ will be replaced by the box $\Box_{\mu(\{N_{i_j}\}_K)}$, which is the same box as the one that replaces $\{N_{i_j}\}_K$ in M when the pattern of M is computed. Hence $M \cong M'$, and therefore, since soundness is assumed, and *B-Keys*(M') is not cyclic in M', we have that $[\![M]\!]_\Phi \approx [\![M']\!]_\Phi$.

In order to prove that (i) follows from (ii), consider two equivalent valid expressions M and N, such that $M \cong N$. Then, by definition, there exists a bijection σ on **Keys** (preserving $\equiv_{\mathbf{K}}$) such that *pattern*$(M) =$ *pattern*$(N\sigma)$. This means that the "boxes" occurring in *pattern*(M) must occur in *pattern*$(N\sigma)$ and vice-versa. Also, the subexpressions of *pattern*(M) and of *pattern*$(N\sigma)$ outside the boxes must agree as well. Hence,

$$\textit{R-Keys}(M) = \textit{R-Keys}(N\sigma) = \textit{R-Keys}(N)\sigma.$$

Let $L_1, L_2, \ldots, L_b$ ($L_i \neq L_j$ if $i \neq j$) denote the keys in *B-Keys*(M), and let L'_1, L'_2, …, $L'_{b'}$ ($L'_i \neq L'_j$ if $i \neq j$) denote the keys in *B-Keys*$(N)\sigma$. *B-Keys*(M) and *B-Keys*(N) (and therefore *B-Keys*$(N\sigma)$ as well) are not cyclic by hypothesis, so without loss of generality, we can assume that the L_is and the L'_is are numbered in such a way that L_i encrypts L_j (respectively, L'_i encrypts L'_j) only if $i < j$ (for a more detailed argument about this, see [3]; intuitively this means that those keys in *B-Keys*(M) that are deeper in M have a higher number.

Consider now the set of expressions that are subexpressions of M or N and have the form $\{M'\}_{L_i}$ or $\{N'\}_{L'_i}$, and also, the set S. Condition (ii) then provides a set with elements of the form $C_{\mu(\{M'\}_{L_i})}$ and $C_{\mu(\{N'\}_{L'_i})}$.

Let $M_0 = M$. Let M_1 be the expression obtained from M_0 by replacing all subexpressions in M_0 of the form $\{M'\}_{L_1}$ by $C_{\mu(\{M'\}_{L_1})}$ given by the assumption. Let then M_i, $i \geq 2$, be the expression obtained from M_{i-1} by replacing all subexpressions in M_{i-1} of the form $\{M'\}_{L_i}$ by $C_{\mu(\{M'\}_{L_i})}$. We do this for all $i \leq b$ and it is easy to see that in M_b replacing the subexpressions of the form $C_{\mu(\{M'\}_{L_i})}$ by $\Box_{\mu(\{M'\}_{L_i})}$ for all i, we obtain *pattern*(M).

Note that in M_{i-1}, L_i can only occur as an encrypting key. The reason for this is that if L_i is a subexpression of M, then it has to be encrypted with some non-recoverable key, otherwise L_i would be recoverable; moreover, it has to be encrypted with some key in *B-Keys*(M) because a subexpression of M is either recoverable or ends up in a box when we construct *pattern*(M). Now, the element in *B-Keys*(M) that encrypts L_i has to be an L_j with $j < i$. But, all subexpressions in M of the form $\{M'\}_{L_j}$ were already replaced by $C_{\mu(\{M'\}_{L_j})}$ when we constructed M_j. According to the properties listed in proposition 48, L_i may only appear in $C_{\mu(\{M'\}_{L_j})}$ as the encrypting key, and then $L_i = L_j$, a contradiction. So L_i cannot appear in M_{i-1} in any other place than an encrypting key. Observe as well, that *R-Keys*$(M_i) =$ *R-Keys*(M).

From assumption (ii), it follows then that $[\![M_{i-1}]\!]_{\Phi} \approx [\![M_i]\!]_{\Phi}$, for all i, $1 \le i \le b$. Hence,

$$[\![M]\!]_{\Phi} = [\![M_0]\!]_{\Phi} \approx [\![M_b]\!]_{\Phi}. \tag{3}$$

Carrying out the same process for $N\sigma$ through $(N\sigma)_0$, $(N\sigma)_1, \ldots, (N\sigma)_{b'}$ we arrive at

$$[\![(N\sigma)]\!]_{\Phi} = [\![(N\sigma)_0]\!]_{\Phi} \approx [\![(N\sigma)_{b'}]\!]_{\Phi}. \tag{4}$$

Since we supposed that $M \cong N$, that is, *pattern*$(M) =$ *pattern*$(N\sigma)$, and therefore $M_b =$ *pattern*(M) and $(N\sigma)_{b'} =$ *pattern*$(N\sigma)$, we have

$$[\![M_b]\!]_{\Phi} = [\![(N\sigma)_{b'}]\!]_{\Phi}. \tag{5}$$

Then, it is clearly true that

$$[\![N]\!]_{\Phi} = [\![N\sigma]\!]_{\Phi} \tag{6}$$

because permuting the keys in N does not have any effect in the distributions. Putting together Equations (3), (4), (5) and (6) the soundness result follows: $[\![M]\!]_{\Phi} \approx [\![N]\!]_{\Phi}$.

□

Remark 55. The reader might ask why we do not have a similar general theorem for key cycles and KDM-like security. The reason is that this general soundness theorem tells us under which conditions the several steps of the Abadi-Rogaway hybrid argument can be carried out. One of the conditions is that by doing one step of replacement, we must obtain equivalent interpretations, provided that we have the appropriate security notion. However, in our theorem using KDM security to solve the key cycles issue, there is only one step of replacement! All the replacements of undecryptable terms are done at once. Therefore, in a general theorem (without assuming a specific security level), the condition of the theorem would have to be exactly what we would want to prove, and that makes no sense.

We illustrate this general theorem by applying it to three detailed examples. First, we consider encryption schemes which may reveal the length of the plaintext, but which conceal whether or not two ciphertexts were created using the same key. In the terminology of Abadi and Rogaway [3] these are known as ‘type-1’ encryption schemes:

Definition 56 (Type-1 Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be symmetric encryption scheme. We say that the encryption-scheme is* type-1 secure *if no PPT adversary* A *can distinguish the pair of oracles* $(\mathcal{E}(k, \cdot), \mathcal{E}(k', \cdot))$ *and* $(\mathcal{E}(k, 0^{|\cdot|}), \mathcal{E}(k, 0^{|\cdot|}))$ *as* k *and* k' *are* *independently generated, that is, for all PPT adversaries* A*:*

$$\Pr\left[k,k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot),\mathcal{E}(k',\cdot)}(1^\eta)=1\right] - \Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,0^{|\cdot|}),\mathcal{E}(k,0^{|\cdot|})}(1^\eta)=1\right] \leq \mathrm{neg}\,(\eta)\,.$$

In this example (which ends with Corollary 58) we use our general soundness theorem to prove soundness for these schemes.

Example 57 (Type-1 Soundness). Let $\mathbf{Exp}_{\mathcal{V}}$ be the set of elements in $M \in \mathbf{Exp}$ for which *B-Keys*(M) is not cyclic. The equivalence relation $\equiv_1$ is as in Example 35, which is proper by Example 45; the equivalence relation $\equiv_{\mathbf{K}}$ is trivial here, all keys are equivalent. The elements $\mu \in \mathcal{Q}_{\mathbf{Enc}}$ are in one-to-one correspondence with the possible length, so the patterns that we obtain this way are the same that we defined in Example 42, and the equivalence of expressions is $\cong_1$ that is defined in the same example. In order to see that condition (ii) of the general soundness theorem is satisfied for type-1, we use the following equivalent definition of type-1 secure encryption schemes: an encryption-scheme is *type-1 secure* if no *PPT* adversary A can distinguish the pair of oracles $(\mathcal{E}(k,\cdot,\cdot,0),\mathcal{E}(k',\cdot,\cdot,0))$ and $(\mathcal{E}(k,\cdot,\cdot,1),\mathcal{E}(k,\cdot,\cdot,1))$ as k and k' are independently generated, that is, for all *PPT* adversaries A:

$$\Pr\left[k,k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot,\cdot,0),\mathcal{E}(k',\cdot,\cdot,0)}(1^\eta)=1\right] - \Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot,\cdot,1),\mathcal{E}(k,\cdot,\cdot,1)}(1^\eta)=1\right] \leq \mathrm{neg}\,(\eta)$$

where oracle $\mathcal{E}(k,\cdot,\cdot,0)$, upon the submission of two messages with equal length encrypts the first, while oracle $\mathcal{E}(k,\cdot,\cdot,1)$ encrypts the second.

To show that condition (ii) of Theorem 54 holds, we first have to choose $\{C_\nu\}_{\nu\in\mu(\mathfrak{C})}$ for a given set $\mathfrak{C} = \{\{N_i\}_{L_i}\}_{i=1}^n$. We can choose any family $\{C_\nu\}_{\nu\in\mu(\mathfrak{C})}$ such that all the C_ν are encrypted with the same key, let us call it L_0, that is not present in any of the $\{N_i\}_{L_i}$ neither in M. This is possible because, as it is easy to check, $\nu_{\mathrm{key}} = \mathbf{Keys}$ for all $\nu \in \mathcal{Q}_{\mathbf{Enc}}$. Then, let M be as in condition (ii) of Theorem 54. We need to show that if $\{\{N_{i_j}\}_L\}_{j=1}^l \subseteq \mathfrak{C}$ and if we denote by M' the expression obtained from M by replacing each $\{N_{i_j}\}_L$ with $C_{\mu(\{N_{i_j}\}_L)}$, then $[\![M]\!]_\Phi \approx [\![M']\!]_\Phi$.

Suppose that $[\![M]\!]_\Phi \not\approx [\![M']\!]_\Phi$. This means that there is an adversary A that is able to distinguish the two distributions, that is $\Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta,x)=1] - \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta,x)=1]$ is a non-negligible function of η. We will show that this contradicts type-1 security. To this end, we construct an adversary that can distinguish between the two pairs of oracles above. This adversary is the following probabilistic algorithm that has access to the oracles f and g:

algorithm $\mathsf{B}^{f,g}(1^\eta, M)$
  for $K \in$ *Keys*$(M) \setminus \{L, L_0\}$ **do** $\tau(K) \longleftarrow \mathcal{K}(1^\eta)$
  $y \longleftarrow \mathrm{CONVERT2}(M)$
  $b \longleftarrow \mathsf{A}(1^\eta, y)$
  return b

algorithm CONVERT2(N)
    if $N = K$ where $K \in$ **Keys then**
        return $\tau(K)$
    if $N = B$ where $B \in$ **Blocks then**
        return B
    if $N = (M_1, M_2)$ **then**
        $x \longleftarrow \mathrm{CONVERT2}(M_1)$
        $y \longleftarrow \mathrm{CONVERT2}(M_2)$
        return $[x, y]$
    if $N = \{M_1\}_L$ **then**
        $x \longleftarrow \mathrm{CONVERT2}(M_1)$
        $y \longleftarrow \mathrm{CONVERT2}(M_\nu)$ (where $C_{\mu(\{M_1\}_L)} = \{M_\nu\}_{L_0}$)
        $z \longleftarrow f(x, y)$
        return z
    if $N = \{M_1\}_{L_0}$ **then**
        $x \longleftarrow \mathrm{CONVERT2}(M_1)$
        $y \longleftarrow g(x, x)$
        return y
    if $N = \{M_1\}_K$ $(K \notin \{L, L_0\})$ **then**
        $x \longleftarrow \mathrm{CONVERT2}(M_1)$
        $y \longleftarrow \mathcal{E}(\tau(K), x)$
        return y

Note that the algorithm CONVERT2 does almost the same as the algorithm CONVERT of Figure 1, except that while CONVERT carries out all the necessary encryptions, CONVERT2 makes the oracles carry out the encryptions for L and L_0. It is easy to see that if the pair of oracles (f, g) is $(\mathcal{E}(k, \cdot, \cdot, 0), \mathcal{E}(k', \cdot, \cdot, 0))$, then CONVERT2$(M)$ is a random sample from $[\![M]\!]_{\Phi_\eta}$, whereas if the pair of oracles is $(\mathcal{E}(k, \cdot, \cdot, 1), \mathcal{E}(k, \cdot, \cdot, 1))$, then CONVERT2$(M)$ is a random sample from $[\![M']\!]_{\Phi_\eta}$. Thus, $\Pr[k, k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,\cdot,\cdot,0), \mathcal{E}(k',\cdot,\cdot,0)}(1^\eta, M) = 1] = \Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$ and $\Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,\cdot,\cdot,1), \mathcal{E}(k,\cdot,\cdot,1)}(1^\eta, M) = 1] = \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$. But, according to our assumption, $[\![M]\!]_\Phi$ and $[\![M']\!]_\Phi$ can be distinguished, that is, $\Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1] - \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$ is a non-negligible function of η and so, there is an adversary $\mathsf{B}^{f,g}(1^\eta, \cdot)$ such that

$$\Pr[k, k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,\cdot,\cdot,0), \mathcal{E}(k',\cdot,\cdot,0)}(1^\eta, M) = 1] - \Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,\cdot,\cdot,1), \mathcal{E}(k,\cdot,\cdot,1)}(1^\eta, M) = 1]$$

is also a non-negligible function of η. This implies that our scheme cannot be type-1 secure, which contradicts the assumption. Hence, we cannot have $[\![M]\!]_\Phi \not\approx [\![M']\!]_\Phi$ and so condition (ii) of the general soundness theorem is satisfied, which implies that soundness holds for the type-1 case. To summarize, we have that

Corollary 58 (Type-1 Soundness). *Let Π be a type-1 secure encryption scheme such that for each η security parameter, if $k, k' \longleftarrow \mathcal{K}(1^\eta)$, then $|k| = |k'|$, and for any m*

plaintext, $|\mathcal{E}(k,m,w)| = |\mathcal{E}(k',m,w')|$ *for all* $w, w' \longleftarrow$ **coins***. Then, if the length-function satisfies only the equalities defined in Definition 19, then for any* M *and* N *expressions such that B-Keys*(M) *and B-Keys*(N) *are not cyclic in* M *and* N *respectively,* $M \cong_1 N$ *implies* $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$*.*

Otherwise, for arbitrary length-function ℓ *(that is, one satisfying possible more equations), if for all pairs of expressions* M *and* N*,* $\ell(M) = \ell(N)$ *implies that the binary length of* $[\![M]\!]_{\Phi_\eta}$ *is the same as the binary length of* $[\![N]\!]_{\Phi_\eta}$ *for each security parameter* η*, then for any* M *and* N *expressions,* $M \cong_1 N$ *implies* $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$*.*

Having considered the leakage of plaintext-length in the previous example, we now turn to another kind: whether or not two ciphertexts share a key. Schemes which conceal plaintext-length but reveal this information are called 'type-2' in the terminology of Abadi and Rogaway, or are 'message-concealing' and 'length-concealing' but may be 'which-key revealing.' For this type of encryption, no adversary should be able to tell whether a ciphertext contains a (possibly long) plaintext or the single-bit plaintext 0:

Definition 59 (Type-2 Security). *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be symmetric encryption scheme. We say that the encryption-scheme is* type-2 secure *if no PPT adversary* A *can distinguish the oracles* $\mathcal{E}(k,\cdot)$ *and* $\mathcal{E}(k,0)$ *as* k *is randomly generated, that is, for all PPT adversaries* A*:*

$$\Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot)}(1^\eta) = 1\right] - \Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,0)}(1^\eta) = 1\right] \leq \mathrm{neg}\,(\eta)\,.$$

In the next example (which ends with Corollary 62) we apply Theorem 54 to type-2 encryption schemes.

Example 60 (Type-2 Soundness). Let $\mathbf{Exp}_{\mathcal{V}}$ be the set of elements in $M \in \mathbf{Exp}$ for which *B-Keys*(M) is not cyclic; the equivalence relation $\equiv_2$ is as in Example 36, which is proper as shown in Example 44; the equivalence relation $\equiv_{\mathbf{K}}$ is trivial here, all keys are equivalent. The elements $\mu \in \mathcal{Q}_{\mathbf{Enc}}$ are in one-to-one correspondence with the keys, so we can say $\mathcal{Q}_{\mathbf{Enc}} \equiv \mathbf{Keys}$, and thus the boxes are labeled with keys. The patterns *pattern*$_2$ and the equivalence of expressions $\cong_2$ were defined in Example 41. Then for a set $\mathfrak{C} = \{\{N_i\}_{L_i}\}_{i=1}^n$ as in condition (ii) of the Theorem 54, we can take $C_{L_i} := \{0\}_{L_i}$, and the condition is satisfied, because the following proposition holds:

Proposition 61. *Let* $M \in \boldsymbol{Exp}_{\mathcal{V}}$ *and* $L \in Keys(M)$*. Let* $\{M_1\}_L, \{M_2\}_L, \ldots, \{M_l\}_L \sqsubseteq M$*, be such that that* L *does not occur anywhere else in* M*. Then, denoting by* M' *the expression that is obtained from* M *by replacing each* $\{M_i\}_L$ *that is not contained in any other* M_j *(*$j \neq i$*) with* $\{0\}_L$*, we have that* $[\![M]\!]_\Phi \approx [\![M']\!]_\Phi$ *whenever the expressions are interpreted with a type-2 encryption scheme.*

Proof. We can assume, without loss of generality, that $\{M_i\}_L$ is a subexpression of $\{M_j\}_L$ only if $i < j$. Suppose that $[\![M]\!]_\Phi \not\approx [\![M']\!]_\Phi$. This means that there is an adversary A that is able to distinguish the two distributions, that is $\Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1] - \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$ is a non-negligible function of η. We will show that this contradicts type-2 security. To this end, we construct an adversary that can distinguish between oracle $\mathcal{E}(k,\cdot)$ and $\mathcal{E}(k,0)$. This adversary is the following probabilistic algorithm that has access to the oracle f:

algorithm $\mathsf{B}^f(1^\eta, M)$
  for $K \in \mathit{Keys}(M) \setminus \{L\}$ **do** $\tau(K) \longleftarrow \mathcal{K}(1^\eta)$
  $y \longleftarrow \mathrm{CONVERT2}(M)$
  $b \longleftarrow \mathsf{A}(1^\eta, y)$
  return b

algorithm CONVERT2(N)
  if $N = K$ where $K \in$ **Keys then**
    return $\tau(K)$
  if $N = B$ where $B \in$ **Blocks then**
    return B
  if $N = (N_1, N_2)$ **then**
    $x \longleftarrow \mathrm{CONVERT2}(N_1)$
    $y \longleftarrow \mathrm{CONVERT2}(N_2)$
    return $[x, y]$
  if $N = \{N_1\}_L$ **then**
    $x \longleftarrow \mathrm{CONVERT2}(N_1)$
    $y \longleftarrow f(x)$
    return y
  if $N = \{N_1\}_K$ $(K \neq L)$ **then**
    $x \longleftarrow \mathrm{CONVERT2}(N_1)$
    $y \longleftarrow \mathcal{E}(\tau(K), x)$
    return y

Note that the algorithm CONVERT2 does almost the same as the algorithm CONVERT of Figure 1, except that while CONVERT carries out all the necessary encryptions, CONVERT2 makes the oracles carry out the encryptions for L. It is easy to see that if the oracle f is $\mathcal{E}(k, \cdot)$, then CONVERT2(M) is a random sample from $[\![M]\!]_{\Phi_\eta}$, whereas if the oracle is $\mathcal{E}(k, 0)$, then CONVERT2(M) is a random sample from $[\![M']\!]_{\Phi_\eta}$. Thus, $\Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,\cdot)}(1^\eta, M) = 1] = \Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$ and also $\Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{B}^{\mathcal{E}(k,0)}(1^\eta, M) = 1] = \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$. But, according to our assumption, $[\![M]\!]_\Phi$ and $[\![M']\!]_\Phi$ can be distinguished, that is, $\Pr[x \longleftarrow [\![M]\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1] - \Pr[x \longleftarrow [\![M']\!]_{\Phi_\eta} : \mathsf{A}(1^\eta, x) = 1]$ is a non-negligible function of η and so, there is an adversary $\mathsf{B}^f(1^\eta, \cdot)$ that can distinguish the oracles $\mathcal{E}(k, \cdot)$ and $\mathcal{E}(k, 0)$, for randomly generated keys k. This implies that our scheme cannot be type-2 secure, which contradicts the assumption. Hence, we cannot have $[\![M]\!]_\Phi \not\approx [\![M']\!]_\Phi$. □

Hence, condition (ii) of the general soundness theorem is satisfied, which implies that soundness holds also for the type-2 case.

Corollary 62 (Type-2 Soundness). *Let Π be a type-2 secure encryption scheme, and let M and N be two valid expressions such that B-Keys(M) and B-Keys(N) are not cyclic in M and N respectively. Then, $M \cong_2 N$ implies $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$.*

In our last example (which ends with Corollary 65) we apply Theorem 54 to the One-Time Pad.

Example 63 (Soundness for One-Time Pad). We have introduced the formalism for OTP in Examples 30, 33, 37, 41, 44. Then for a set $\mathfrak{C} = \{\{N_i\}_{L_i}\}_{i=1}^n$ as in condition (ii) of Theorem 54, take $C_{L_i} := \{0^{l(L_i)-3}\}_{L_i}$. It is not hard to check that within this setting, condition (ii) of Theorem 54 is satisfied, which is an immediate consequence of the following proposition:

Proposition 64. *Let $M \in \boldsymbol{Exp}_{\mathrm{OTP}}$ and $K_0 \in Keys(M)$. Let $\{M_0\}_{K_0} \sqsubseteq M$, be such that K_0 does not occur anywhere else in M. Then, denoting by M' the expression that is obtained from M by replacing $\{M_0\}_{K_0}$ with $\{0^{l(K_0)-3}\}_{K_0}$, we have that $[\![M]\!]_\Phi \approx [\![M']\!]_\Phi$ when Φ is the interpretation for OTP.*

Proof. The basic properties of the OTP ensure that $\Phi(\{M_0\}_{K_0})$ is evenly distributed over the set of $l(K_0)$ long strings ending with 110, no matter what M_0 is. So the distribution of $\Phi(\{M_0\}_{K_0})$ agrees with the distribution of $\Phi(\{0^{l(K_0)-3}\}_{K_0})$. Also, since K_0 is assumed not to occur anywhere else, $\Phi_M(K_0)$ is independent of the interpretation of the rest of the expression M, and therefore, $\Phi(\{M_0\}_{K_0})$ and $\Phi(\{0^{l(K_0)-3}\}_{K_0})$ are both independent of the interpretation of the rest of the expression. Hence, replacing $\Phi(\{M_0\}_{K_0})$ with $\Phi(\{0_{l(K_0)-3}\}^{K_0})$ will not affect the distribution. $\square$

Hence, condition (ii) of the general soundness theorem is satisfied, which implies that soundness holds also for the OTP case.

Corollary 65 (OTP Soundness). *Let M and N be two valid expressions in $\boldsymbol{Exp}_{\mathrm{OTP}}$ such that B-Keys(M) and B-Keys(N) are not cyclic in M and N respectively. Then, $M \cong_{OTP} N$ implies that $[\![M]\!]_\Phi$ and $[\![N]\!]_\Phi$ have the same probability distributions.*

5.5 Parsing Process

The technique that we present in this section will be very useful in the course of proving our completeness results. The idea can be summarized as follows: given a sample element $x \longleftarrow [\![M]\!]_\Phi$, x is built from blocks and randomly generated keys which are paired and encrypted. Some of the keys that were used for encryption when x was built might be explicitly contained in x, and in this case, using these keys, we can decrypt those ciphers that were encrypted with these revealed keys. The problem is though, that looking at x, it might not be possible to tell where blocks, keys, ciphers and pairs are in the string of bits, since we did not assume in general that we tag strings as we did for OTP. However, and we will exploit this fact repeatedly in our proofs, if we know that x was sampled from $[\![M]\!]_\Phi$ for a fixed, known expression M, then by looking at M, we can find in x the locations of blocks, keys, ciphers and pairs, and we can also tell from M, where the key decrypting a certain cipher is located. We present a machinery that, using the form of an expression M, extracts from an $x \longleftarrow [\![M]\!]_\Phi$ everything that is possible via decryption and depairing, and distributes the extracted elements over a special Cartesian product of copies of $\overline{\mathtt{strings}}$.

Throughout this section, we assume given a logic $\Delta = (\mathbf{Exp}_\mathcal{V}, \equiv_\mathbf{K}, \equiv_\mathbf{C})$, a general symmetric encryption scheme $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$, and an interpretation Φ for Π. For $i = 1, 2$, let $[\cdot,\cdot]_i^{-1} := \pi^i \circ [\cdot,\cdot]^{-1}$, where $[\cdot,\cdot]$ is the computational pairing, defined in Section 5.1. Let

$$\mathcal{EXP} ::= \overline{\mathbf{strings}} \mid (\mathcal{EXP}, \mathcal{EXP}) \mid \{\mathcal{EXP}\}_{\overline{\mathbf{strings}}}$$

For example, $((\{a\}_b, \{c\}_d), ((e, \{(\{f\}_g, \{b\}_f)\}_f), \{f\}_e))$ is an element of $\mathcal{EXP}$ if $a, b, c, d, e, f, g \in \overline{\mathbf{strings}}$.

Definition 66. *Let $M \in$ **Exp**$_{\mathcal{V}}$ and $\{L_1, \ldots, L_n\}$ an enumeration of R-Keys(M) in the order they can be recovered. For $1 \leq i \leq n$, $N \sqsubseteq M$, and $y \in \overline{\mathbf{strings}}$, we define $\mathcal{B}_i^{N,M,y} : \overline{\mathbf{strings}} \to \mathcal{EXP}$, and $\mathcal{G}^{L_i,M} : \overline{\mathbf{strings}} \to \overline{\mathbf{strings}}$ in the following way:*

$$\mathcal{B}_i^{K,M,y}(x) := x \quad \text{for } K \in \textit{\textbf{Keys}}, \qquad \mathcal{B}_i^{B,M,y}(x) := x \quad \text{for } B \in \textit{\textbf{Blocks}}$$

$$\mathcal{B}_i^{(M_1,M_2),M,y}(x) := \left([\cdot,\cdot]_1^{-1}(x), [\cdot,\cdot]_2^{-1}(x)\right)$$

$$\mathcal{B}_i^{\{N\}_K,M,y}(x) := \begin{cases} \left\{\mathcal{B}_i^{N,M,y}(\mathcal{D}(\mathcal{G}^{K,M}(y), x))\right\}_{\mathcal{G}^{K,M}(y)} & \text{for } K \in \{L_1, \ldots, L_{i-1}\}, \\ x & \text{otherwise}. \end{cases}$$

Let $\mathcal{B}_i^M(x) := \mathcal{B}_i^{M,M,x}(x)$, and $\mathcal{B}^M : \overline{\mathbf{strings}} \to \mathcal{EXP}$ with $\mathcal{B}^M := \mathcal{B}_{n+1}^M$. The function $\mathcal{B}_i^M(x)$ parses x according to M until everything that can be decrypted with the keys $L_1, \ldots, L_{i-1}$ is decrypted, and returns an element of $\mathcal{EXP}$. This has to reveal the string corresponding to L_i, and let $\mathcal{G}^{L_i,M}(x)$ be such string. $\mathcal{B}^M$ parses everything that is decryptable. If something is not in the domain of the corresponding operation, then the algorithm outputs an error message $\perp$. Let $\mathcal{T}(M)$ be the image of $\mathcal{B}^M$.

The following lemma essentially claims that if the interpretation is such that conditions (i) and (ii) below hold, then for any two valid expressions M and N, the distribution of $\mathcal{B}^M(x)$, where x is sampled from $[\![M]\!]_\Phi$ (let $\mathcal{B}^M([\![M]\!]_\Phi)$ denote this distribution), is indistinguishable from the distribution of $\mathcal{B}^N(y)$, where y is sampled from $[\![N]\!]_\Phi$ whenever $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$.

For a function f on $\overline{\mathbf{strings}}$, let $f([\![M]\!]_\Phi)$ denote the probability distribution of $f(x)$ as x is sampled from $[\![M]\!]_\Phi$.

Lemma 67. *Let $\Delta = ($**Exp**$_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be a formal logic for symmetric encryption, and Φ be an interpretation of **Exp**$_{\mathcal{V}}$ in $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$. Suppose that this realization satisfies the following properties for any $K, K', K'' \in$ **Keys**, $B_1 \neq B_2 \in$ **Blocks**, $M, M', N \in$ **Exp**$_{\mathcal{V}}$:*

(i) no pair of $[\![K]\!]_\Phi$, $[\![B_1]\!]_\Phi$, $[\![B_2]\!]_\Phi$, $[\![(M,N)]\!]_\Phi$, $[\![\{M'\}_{K'}]\!]_\Phi$ are equivalent with respect to $\approx$; that is, keys, blocks, pairs, ciphers are distinguishable.
(ii) If $[\![(K, \{M\}_K)]\!]_\Phi \approx [\![(K'', \{M'\}_{K'})]\!]_\Phi$, then $K' = K''$.

Let M and N be two valid formal expressions. If $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$, then $\mathcal{B}^M = \mathcal{B}^N$ and $\mathcal{B}^M([\![M]\!]_\Phi) \approx \mathcal{B}^N([\![N]\!]_\Phi)$.

Proof. We just sketch the proof, a more detailed version can be found in [14]. Let M and N be expressions such that $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$. Since we assumed condition (i) and since the equivalence $\approx$ is assumed to be invariant under depairing, the pairs that are not encrypted in M and in N must be in the same positions, and so $\mathcal{B}_1^M = \mathcal{B}_1^N$ must hold. Since these are obtained by repeated application of the inverse of the pairing function, projecting and coupling, $\mathcal{B}_1^M([\![M]\!]_\Phi) \approx \mathcal{B}_1^N([\![N]\!]_\Phi)$ by our assumptions on indistinguishability in Section 5.1.

Let *R-Keys*$(M) = \{L_1, \ldots, L_{c(M)}\}$ be an enumeration of all recoverable keys in M such that they can be recovered in this order. There must be a position in the image of $\mathcal{B}_1^M$ that corresponds to L_1, that gives us the function $\mathcal{G}^{L_1,M}$. But applying $\mathcal{G}^{L_1,M}$ to $[\![N]\!]_\Phi$ must also reveal a key because $\mathcal{G}^{L_1,M}([\![M]\!]_\Phi) \approx \mathcal{G}^{L_1,M}([\![N]\!]_\Phi)$, so the corresponding entry in N must also be a recoverable key; let's call it L_1'. Clearly, $\mathcal{G}^{L_1,M} = \mathcal{G}^{L_1',N}$. $\mathcal{B}_2^M$ can now be obtained as a composition of $\mathcal{B}_1^M$ with a function that decrypts all entries of $\mathcal{B}_1^M(x)$ (using $\mathcal{G}^{L_1,M}(x)$) that correspond to encryption terms in M with encrypting key L_1, and further applications of the inverse of pairing, projection and coupling again. As $\mathcal{B}_1^M([\![M]\!]_\Phi) \approx \mathcal{B}_1^N([\![N]\!]_\Phi)$, the pairs again must be in the same positions here, and because of condition (ii) of the Lemma, those encryptions that are done with $\mathcal{G}^{L_1,M}(x) = \mathcal{G}^{L_1',N}(x)$ must also be in the same positions. Therefore, $\mathcal{B}_2^M = \mathcal{B}_2^N$ and $\mathcal{B}_2^M([\![M]\!]_\Phi) \approx \mathcal{B}_2^N([\![N]\!]_\Phi)$. Then again, $\mathcal{G}^{L_2,M}$ can be identified, and an $L_2' \in$ *R-Keys*(N) such that $\mathcal{G}^{L_2,M} = \mathcal{G}^{L_2',N}$ and so on until all recoverable keys are recovered and everything that was decryptable has been decrypted. $\square$

5.6 Completeness

We finally present our completeness result. First, note that the theorem below does not mention key cycles. Secondly, note that Condition (i) requires that different types of objects, blocks, keys, pairs and encryption terms should be distinguishable to achieve completeness; this can be ensured by tagging each object with its type, as suggested in [3]. Thirdly, Condition (ii) (which we call *weak confusion-freeness*) is equivalent to the property of weak key-authenticity introduced by Horvitz and Gligor [35] in the case of type-0 schemes. This property essentially means that decrypting with the wrong key should be detectable in a probabilistic sense.

The proof consists of two separate parts. In the first, it is shown that conditions (i) and (ii) imply that if M and N are valid expressions and $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$, then there is a key-renaming function σ, such that apart from the boxes, everything else in the patterns of M and $N\sigma$ is the same, and the boxes in the two patterns must be in the same positions. Moreover, condition (iii) implies that picking any two boxes of the pattern of $N\sigma$, there is a key-renaming function σ_1 such that applying it to the indexes of these boxes, we obtain the corresponding boxes in the pattern of M. Then the theorem follows, if we can prove that using these pairwise equivalences of boxes, we can construct a σ' that leaves untouched the recoverable keys of $N\sigma$ (all the keys outside the boxes), and that maps the indexes of all the boxes of $N\sigma$ into the indexes of the boxes of M.

Theorem 68 (Completeness). *Let $\Delta = (\boldsymbol{Exp}_{\mathcal{V}}, \equiv_{\mathbf{K}}, \equiv_{\mathbf{C}})$ be a formal logic for symmetric encryption such that $\equiv_{\mathbf{C}}$ is proper. Let Φ be an interpretation in $\Pi = (\{\mathcal{K}_i\}_{i\in I}, \mathcal{E}, \mathcal{D}, \approx)$. Completeness for Φ holds, if and only if the following conditions are satisfied: For any $K, K', K'' \in$ **Keys**, $B_1 \neq B_2 \in$ **Blocks**, $M, M', N \in \boldsymbol{Exp}_{\mathcal{V}}$,*

(i) no pair of $[\![K]\!]_\Phi$, $[\![B_1]\!]_\Phi$, $[\![B_2]\!]_\Phi$, $[\![(M,N)]\!]_\Phi$, $[\![\{M'\}_{K'}]\!]_\Phi$ is equivalent with respect to $\approx$; that is, keys, blocks, pairs, encryption terms are distinguishable,

(ii) if $[\![(K, \{M\}_K)]\!]_\Phi \approx [\![(K'', \{M'\}_{K'})]\!]_\Phi$, then $K' = K''$,

(iii) for any two pairs $(\{M_1\}_{L_1}, \{M_2\}_{L_2})$ and $(\{N_1\}_{L_1'}, \{N_2\}_{L_2'})$ of valid encryption terms, we have that $[\![(\{M_1\}_{L_1}, \{M_2\}_{L_2})]\!]_\Phi \approx [\![(\{N_1\}_{L_1'}, \{N_2\}_{L_2'})]\!]_\Phi$ implies $(\{M_1\}_{L_1}, \{M_2\}_{L_2}) \cong (\{N_1\}_{L_1'}, \{N_2\}_{L_2'})$.

Proof. The only if part is trivial. In order to prove the if part, consider two expressions M and N such that $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$. By condition (i) and (ii), Lemma 67 is applicable, so, $\mathcal{B}^M([\![M]\!]_\Phi) \approx \mathcal{B}^N([\![N]\!]_\Phi)$, and $\mathcal{T}(M) = \mathcal{T}(N)$.

In each entry of $\mathcal{T}(M)$ and $\mathcal{T}(N)$, the distribution corresponds either to the interpretation of a key, a block, or an undecryptable cipher (*i.e.*one that corresponds to a box). Naturally, the same blocks must be in the same positions of $\mathcal{T}(M)$ and $\mathcal{T}(N)$, using the fact that the distributions of $\mathcal{B}^M([\![M]\!]_\Phi)$ and $\mathcal{B}^N([\![N]\!]_\Phi)$ are indistinguishable and condition (i). Hence, the patterns of M and N contain the same blocks in the same positions. Also by indistinguishability of $\mathcal{B}^M([\![M]\!]_\Phi)$ and $\mathcal{B}^N([\![N]\!]_\Phi)$ and condition (i), the entries in $\mathcal{T}(M)$ and $\mathcal{T}(N)$ containing strings sampled from key generation algorithm must be in the same places. Furthermore, the indistinguishability of $\mathcal{B}^M([\![M]\!]_\Phi)$ and $\mathcal{B}^N([\![N]\!]_\Phi)$ also implies that repetitions of a key generation outcome must occur in the same positions of $\mathcal{T}(M)$ and $\mathcal{T}(N)$. (This is a consequence of the properties of key-generation stated in Definition 28.) Therefore the key symbols in the patterns of M and N change together, so it is possible to rename the recoverable keys of N (with a $\equiv_{\mathbf{K}}$ preserving function σ) so that the keys in the pattern of $N\sigma$ are the same as the keys in the pattern of M.

Finally, indistinguishability of $\mathcal{B}^M([\![M]\!]_\Phi)$ and $\mathcal{B}^N([\![N]\!]_\Phi)$ and condition (i) also imply that undecryptable ciphers occur exactly in the same entries of $\mathcal{T}(M)$ and $\mathcal{T}(N)$. This means that in the patterns of M and N boxes appear in the same positions. This fact together with the conclusions of the previous paragraph implies that, apart from boxes, everything else in the patterns of M and $N\sigma$ must be the same. Replacing N with $N\sigma$, we can assume from now on that the recoverable keys of N and M are identical (i.e. *R-Keys*(M) = *R-Keys*(N)), and that the patterns of M and N are the same.Therefore, we only have to show that there is a key renaming τ that carries the boxes of N into the boxes of M without changing the recoverable keys.

Suppose that there are l boxes altogether in the pattern of M (and hence in the pattern of N). Let $\{M_1\}_{L_1}$, $\{M_2\}_{L_2}$,..., $\{M_l\}_{L_l}$ be the l undecryptable terms in M that turn into boxes (in M) and $\{N_1\}_{L'_1}$, $\{N_2\}_{L'_2}$, ..., $\{N_l\}_{L'_l}$ the corresponding undecryptable terms in N. We denote by μ_i and ν_i the (possibly repeated) equivalence classes of $\{M_i\}_{L_i}$ and $\{N_i\}_{L'_i}$, respectively, with respect to $\equiv_{\mathbf{C}}$. Then, as we said above, we have that for $i, j \le l$, $[\![(\{M_i\}_{L_i}, \{M_j\}_{L_j})]\!]_\Phi \approx [\![(\{N_i\}_{L'_i}, \{N_j\}_{L'_j})]\!]_\Phi$ holds since $\mathcal{B}^M([\![M]\!]_\Phi)$ and $\mathcal{B}^N([\![N]\!]_\Phi)$ are indistinguishable, and thus, by condition (iii), $(\{M_i\}_{L_i}, \{M_j\}_{L_j}) \cong (\{N_i\}_{L'_i}, \{N_j\}_{L'_j})$. By definition of $\cong$, there exists a key-renaming function σ_{ij} such that $(\Box_{\mu_i}, \Box_{\mu_j}) = (\Box_{\sigma_{ij}(\nu_i)}, \Box_{\sigma_{ij}(\nu_j)})$, that is, there exists a key-renaming function σ_{ij} such that $\mu_i = \sigma_{ij}(\nu_i)$ and $\mu_j = \sigma_{ij}(\nu_j)$.

What remains to show is that there exists a single key-renaming function τ that does not change the recoverable keys of M and N (recall, *R-Keys*(M) = *R-Keys*(N)), and that maps the boxes in the pattern of N into the corresponding boxes in the pattern of M.

Firstly, we assumed that $\equiv_{\mathbf{C}}$ is proper, therefore, by Proposition 48, each $\Box_{\nu_i}$ of N has a representative C_i such that C_i does not contain elements of *R-Keys*(N) = *R-Keys*(M). Moreover, for any two different ν_i and ν_j, the only common element of the sets *Keys*(C_i) and *Keys*(C_j) may be the encrypting key, and this only happens if there is a single encrypting key for all elements in ν_i and ν_j. Note that we use C_i to denote the representatives of the equivalence classes of encryption terms in N, that is representatives of ν_i, and not the representatives of equivalence classes (μ_i) of encryption terms in M. Let $U = \bigcup_{i=1}^{l}(\|(\mu_i)_{\text{key}}\| \cup \|(\nu_i)_{\text{key}}\|)$. (The definition of $\|\cdot\|$ is just before Proposition 49.)

Now we can define substitution τ inductively by first defining a sequence τ_k for $k = 1, \ldots, l$. By Proposition 49, considering the sets $S_1 = \big(\bigcup_{i=2}^{l}(\mathit{Keys}(C_i) \cup \|(\mu_i)_{\text{key}}\|) \cup \textit{R-Keys}(N)\big) \setminus \big(\mathit{Keys}(C_1) \cup \|(\mu_1)_{\text{key}}\|\big)$ and U, and the encryption term C_1 it is possible to modify σ_{12} such that the σ'_{12} that we get leaves elements of S_1 untouched, $\sigma'_{12}(\mathit{Keys}(C_1) \setminus \|(\nu_1)_{\text{key}}\|) \cap (S_1 \cup U) = \emptyset$, but it still holds that $\sigma'_{12}(\nu_1) = \sigma_{12}(\nu_1) = \mu_1$. Define $\tau_1 := \sigma'_{12}$.

For the induction step suppose that we have defined τ_k such that:

(a) τ_k is the identity map on $S_k = \big(\bigcup_{i=k+1}^{l}(\mathit{Keys}(C_i) \cup \|(\mu_i)_{\text{key}}\|) \cup \textit{R-Keys}(N)\big) \setminus \big(\bigcup_{i=1}^{k}(\mathit{Keys}(C_i) \cup \|(\mu_i)_{\text{key}}\|)\big)$;
(b) $\tau_k\big(\bigcup_{i=1}^{k}(\mathit{Keys}(C_i) \setminus \|(\nu_i)_{\text{key}}\|)\big) \cap U = \emptyset$; and
(c) $\tau_k(\nu_i) = \mu_i$ for all $i \leq k$.

Clearly, τ_1 satisfies these conditions.

In order to define τ_{k+1}, first check if $C_{k+1} = C_i$ for some $i \leq k$. If so, then clearly $\nu_{k+1} = \nu_i$, and considering $\sigma_{i(k+1)}$, one obtains $\mu_{k+1} = \sigma_{i(k+1)}(\nu_{k+1}) = \sigma_{i(k+1)}(\nu_i) = \mu_i$. Therefore we can define $\tau_{k+1} = \tau_k$, and with that, $\tau_{k+1}(\nu_{k+1}) = \tau_{k+1}(\nu_i) = \mu_i = \mu_{k+1}$. The other conditions follow trivially.

If there is no such i, consider $\sigma_{1(k+1)}$ and $U_k = \bigcup_{i=1}^{l} \tau_k(\mathit{Keys}(C_i)) \cup U$. By Proposition 49, it is possible to alter $\sigma_{1(k+1)}$ into σ' such that:

(i) σ' is the identity map on $S'_k = \big(\bigcup_{i=1}^{l}(\mathit{Keys}(C_i) \cup \tau_k(\mathit{Keys}(C_i))) \cup \textit{R-Keys}(N)\big) \setminus \big(\mathit{Keys}(C_{k+1}) \cup \|(\mu_{k+1})_{\text{key}}\|\big)$;
(ii) $\sigma'(\mathit{Keys}(C_{k+1}) \setminus \|(\nu_{k+1})_{\text{key}}\|) \cap (S_k \cup U_k) = \emptyset$; and
(iii) $\sigma'(\nu_{k+1}) = \mu_{k+1}$.

Our goal is now to combine τ_k and σ'.

We define τ_{k+1} to be equal to τ_k on $\bigcup_{i=1}^{k} \mathit{Keys}(C_i)$, to be equal to σ' on $\mathit{Keys}(C_{k+1})$, to map $\big(\bigcup_{i=1}^{k} \tau_k\ (\mathit{Keys}(C_i)) \cup \sigma'\ (\mathit{Keys}(C_{k+1}))\big) \setminus \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i) \cup \mathit{Keys}(C_{k+1})\big)$ bijectively to $\big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i) \cup \mathit{Keys}(C_{k+1})\big) \setminus \big(\bigcup_{i=1}^{k} \tau_k(\mathit{Keys}(C_i)) \cup \sigma'\ (\mathit{Keys}(C_{k+1}))\big)$ (this part is not uniquely determined), and to be the identity map everywhere else.

If τ_{k+1} is well defined, then it has the following properties:

(a') τ_{k+1} is the identity function in S_{k+1}, since τ_k is the identity in S_k, σ' is the identity in S'_k and $S_{k+1} \subseteq S_k \cap S'_k$;
(b') $\tau_{k+1}\big(\bigcup_{i=1}^{k+1}(\mathit{Keys}(C_i) \setminus \|(\nu_i)_{\text{key}}\|)\big) \cap U = \emptyset$, since for all $K \in \bigcup_{i=1}^{k} \mathit{Keys}(C_i)$, $\tau_{k+1}(K) = \tau_k(K)$ and by (b) $\tau_k\big(\bigcup_{i=1}^{k}(\mathit{Keys}(C_i) \setminus \|(\nu_i)_{\text{key}}\|)\big) \cap U = \emptyset$, and for all $K \in \mathit{Keys}(C_{k+1})$, $\tau_{k+1}(K) = \sigma'(K)$ and by (ii) $\sigma'(\mathit{Keys}(C_{k+1}) \setminus \|(\nu_{k+1})_{\text{key}}\|) \cap (S_k \cup U_k) = \emptyset$; and
(c') $\tau_{k+1}(\nu_i) = \mu_i$ for all $i \leq k+1$ by (c) and (iii).

We now show that τ_{k+1} is a well defined bijection. For that, we have to show that for any key that is changed by both τ_k and σ', that is, $K \in \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big) \cap \mathit{Keys}(C_{k+1})$, we have $\tau_k(K) = \sigma'(K)$, and that for the keys that only τ_k or σ' change, that is, $L \in \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big) \setminus \mathit{Keys}(C_{k+1})$, and $L' \in \mathit{Keys}(C_{k+1}) \setminus \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big)$, we have $\tau_k(L) \neq \sigma'(L')$.

Take $K \in \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big) \cap \mathit{Keys}(C_{k+1})$. Then $K \in \mathit{Keys}(C_{k+1}) \cap \mathit{Keys}(C_i)$ for some $i \leq k$. By construction of the family C_j we have that

$$(\nu_{k+1})_{\mathrm{key}} = (\nu_i)_{\mathrm{key}} = \{K\} = \{L'_{k+1}\} = \{L'_i\}. \tag{7}$$

Then we have that $\sigma_{i(k+1)}(\nu_i) = \mu_i$ and $\sigma_{i(k+1)}(\nu_{k+1}) = \mu_{k+1}$. Combining these with (7) and Proposition 47, we obtain that $\sigma_{i(k+1)}(K) = L_i$ and $\sigma_{i(k+1)}(K) = L_{k+1}$, hence

$$L_i = L_{k+1}. \tag{8}$$

Using again (7), Proposition 47, and the fact that $\tau_k(\nu_i) = \mu_i$ and $\sigma'(\nu_{k+1}) = \mu_{k+1}$, one obtains that $\tau_k(K) = \tau_k(L'_i) = L_i$ and $\sigma'(K) = \sigma'(L'_{k+1}) = L_{k+1}$, which by (8) imply $\tau_k(K) = L_i = L_{k+1} = \sigma'(K)$.

Let us now prove the other case. If $L \in \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big) \setminus \mathit{Keys}(C_{k+1})$ and $L' \in \mathit{Keys}(C_{k+1}) \setminus \big(\bigcup_{i=1}^{k} \mathit{Keys}(C_i)\big)$, then we have to show that $\tau_k(L) \neq \sigma'(L')$. Suppose the contrary, $\tau_k(L) = \sigma'(L')$. Since $L \in \bigcup_{i=1}^{k} \mathit{Keys}(C_i)$, we have $\sigma'(L') = \tau_k(L) \in U_k$, and so $\sigma'(L') \in (S_k \cup U_k)$. Therefore, (ii) gives us that $L' \notin \mathit{Keys}(C_{k+1}) \setminus \|(\nu_{k+1})_{\mathrm{key}}\|$, that by the way L' was chosen implies $L' \in \|(\nu_{k+1})_{\mathrm{key}}\|$. Hence

$$(\nu_{k+1})_{\mathrm{key}} = \{L'\} \text{ and } L'_{k+1} = L'. \tag{9}$$

Using (9), Proposition 47 and the fact that $\sigma'(\nu_{k+1}) = \mu_{k+1}$, one obtains $(\mu_{k+1})_{\mathrm{key}} = \{L_{k+1}\}$, and $\sigma'(L') = \sigma'(L'_{k+1}) = L_{k+1}$. Hence $\tau_k(L) = \sigma'(L') = L_{k+1} \in U$ which implies, by (b), that $L \notin \bigcup_{i=1}^{k}(\mathit{Keys}(C_i) \setminus \|(\nu_i)_{\mathrm{key}}\|)$. This further implies, by the way we selected L, that there is an $i \leq k$ with $L \in \|(\nu_i)_{\mathrm{key}}\|$, that is,

$$L = L'_i \text{ for some } i \leq k \text{ and } (\nu_i)_{\mathrm{key}} = \{L'_i\}. \tag{10}$$

By (10), Proposition 47, and $\tau_k(\nu_i) = \mu_i$, we obtain that $\tau_k(L) = \tau_k(L'_i) = L_i$, and so

$$L_{k+1} = \sigma'(L') = \tau_k(L) = L_i. \tag{11}$$

By the definition of $\sigma_{i(k+1)}$, we have $\sigma_{i(k+1)}(\nu_i) = \mu_i$ and $\sigma_{i(k+1)}(\nu_{k+1}) = \mu_{k+1}$. Combining these with (10), (9) and Proposition 47 we obtain that $\sigma_{i(k+1)}(L'_i) = L_i$ and $\sigma_{i(k+1)}(L'_{k+1}) = L_{k+1}$. Using (11) and the fact that $\sigma_{i(k+1)}$ is a bijection we obtain

$$L'_i = L'_{k+1}. \tag{12}$$

Composing (10), (12) and (9) we obtain that $L = L'_i = L'_{k+1} = L'$, which is a contradiction since we have chosen L and L' from disjoint sets.

Define $\tau := \tau_l$. This τ satisfies the required properties, that is, it leaves the recoverable keys of M and N untouched (as each S_k is disjoint from them), but it maps the boxes of the pattern of N into the corresponding boxes in the pattern of M, and that is what we needed to complete the proof. $\square$

Remark 69. Observe, that condition (iii) of the theorem is trivially satisfied when there is only one box, that is, when all encryption terms are equivalent under $\equiv_{\mathbf{C}}$. Also, if completeness holds for a certain choice of $\equiv_{\mathbf{C}}$, then, if $\equiv'_{\mathbf{C}}$ is such that $M \equiv_{\mathbf{C}} N$ implies $M \equiv'_{\mathbf{C}} N$—*i.e.*when $\equiv'_{\mathbf{C}}$ results fewer boxes—then completeness holds for $\equiv'_{\mathbf{C}}$ as well. Therefore, we can say, that the key to completeness is not to have too many boxes.

Example 70 (Completeness for Type-1 and Type-2 Encryption Schemes). The completeness results for type-1 and type-2 encryption schemes are special cases of the previous theorem, because the formal language we introduced for these schemes is such that $\equiv_{\mathbf{C}}$ is proper, and the conditions of the theorems are analogous. In condition (i) of Corollaries 71 and 73 there is only one block, because our interpretation for computational schemes imply that those are distinguishable.

Corollary 71 (Type-1 Completeness). *Let Π be a type-1 secure encryption scheme . We have that, $[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$ implies $M \cong_1 N$ for any pair of expressions M and N if and only if the following conditions hold: for any $K, K', K'' \in$ **Keys**, $B \in$ **Blocks**, $M, M', N \in$ **Exp**,*

(i) no pair of $[\![K]\!]_\Phi$, $[\![B]\!]_\Phi$, $[\![(M,N)]\!]_\Phi$, $[\![\{M'\}_{K'}]\!]_\Phi$ are equivalent with respect to $\approx$,
(ii) if $[\![(K,\{M\}_K)]\!]_\Phi \approx [\![(K'',\{M'\}_{K'})]\!]_\Phi$, then $K' = K''$, and
(iii) if $[\![\{M\}_K]\!]_\Phi \approx [\![\{M'\}_{K'}]\!]_\Phi$ then $\ell(M) = \ell(M')$.

Condition (iii) above requires that encryption of messages with different length should be detectable. Definition 56 *allows* that encryptions of messages of different length may be detected but does not enforce it. That suffices for soundness, but completeness requires that it should be detectable when ciphertexts contain messages of different lengths. Moreover, there is only hope for completeness, if ℓ is such that it correctly indicates what expressions have interpretations of equal lengths, and which ones have differing lengths. If ℓ is such, that is, if $\ell(M) = \ell(N)$ if and only if the interpretations of M and N have equal lengths (up to negligible probability), then a purely computational condition that implies condition (iii) is the notion of *strictly length revealing*:

Definition 72 (Strictly Length Revealing Scheme). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. We say that the encryption-scheme is* strictly length revealing *if it is type-1 secure, but for any natural n, there exists a PPT adversary A such that*

$$\Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot)}(1^\eta) = 1\right] - \Pr\left[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,0^n)}(1^\eta) = 1\right]$$

is a non-negligible function of η.

For type-2 systems, we have the following corollary:

Corollary 73 (Type-2 Completeness). *Let Π be a type-2 secure encryption scheme. We have that,$[\![M]\!]_\Phi \approx [\![N]\!]_\Phi$ implies $M \cong_2 N$ for any pair of expressions M and N if and only if the following conditions hold: for any $K, K', K'' \in$ **Keys**, $B \in$ **Blocks**, $M, M', N, N' \in$ **Exp**,*

(i) no pair of $[\![K]\!]_\Phi$, $[\![B]\!]_\Phi$, $[\![(M,N)]\!]_\Phi$, $[\![\{M'\}_{K'}]\!]_\Phi$ are equivalent with respect to $\approx$,
(ii) if $[\![(K,\{M\}_K)]\!]_\Phi \approx [\![(K'',\{M'\}_{K'})]\!]_\Phi$, then $K' = K''$,
(iii) if $[\![(\{M\}_K,\{M'\}_K)]\!]_\Phi \approx [\![(\{N\}_{K'},\{N'\}_{K''})]\!]_\Phi$ then $K' = K''$.

The conditions of the theorem are similar to the ones for the type-1 case except for condition (iii). This condition requires that encryption with different keys should be detectable. Definition 59 *allows* that encrypting with different keys may be detectable, but it does not *require* it. That suffices for soundness, but such detection *is* required for completeness. It is easily shown that condition (iii) is implied by the purely computational definition of a *strictly key revealing* encryption scheme:

Definition 74 (Strictly Which-Key Revealing Scheme). *Let $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D})$ be a symmetric encryption scheme. We say that the encryption-scheme is* strictly key revealing *if it is type-2 secure, but there exists a PPT adversary* A *such that*

$$\Pr[k,k' \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot),\mathcal{E}(k',\cdot)}(1^\eta) = 1] - \Pr[k \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathcal{E}(k,\cdot),\mathcal{E}(k,\cdot)}(1^\eta) = 1] \leq \mathrm{neg}\,(\eta)$$

is a non-negligible function of η.

Example 75. Suppose we use the KDM secure encryption scheme that we introduced after Definition 16 for interpretation, along with the concrete pairing function defined in Example 30. It is easy to see that this encryption scheme is both strictly length and strictly which-key revealing. Then the formal model for type-3 systems (boxes indexed with both length and encrypting key) is not only sound, but also complete if we use the following length function defined recursively: $l(B) := \{|B|\}_{\eta\in\mathbb{N}}$, $l(K) := \{2\eta\}_{\eta\in\mathbb{N}}$, $l((M,N)) := \{l(M)_\eta + 2l(N)_\eta + 1\}_{\eta\in\mathbb{N}}$, $l(\{M\}_K) := \{l(M)_\eta + 2\eta\}_{\eta\in\mathbb{N}}$. Then we define $l(M) = l(N)$ to hold iff it there is an η_0 such that for any $\eta > \eta_0$, $l(M)_\eta = l(N)_\eta$ holds. Conditions (i) and (ii) of the theorem are easily shown to hold as the distributions of different types cannot be confused, and the correct decrypting key can easily detected via the appended second part of the key. Condition (iii) holds because of the way we chose our length function along with the strictly length and strictly which-key revealing property.

Example 76 (Completeness for One-Time Pad). The formal logic for OTP that we presented in Examples 33, 37, 41, 44 is such that $\equiv_{\mathbf{C}}$ is proper. Furthermore, condition (i) of Theorem 68 is satisfied due to the tagging we presented in Example 30. Condition (ii) is also satisfied because of the tagging: the reason ultimately is that decrypting with the wrong key will sometimes result invalid endings. Condition (iii) is also satisfied, since the pairs of encryption terms must be encrypted with different keys (in OTP, we cannot use keys twice), and the equivalence $[\![(\{M_1\}_{L_1},\{M_2\}_{L_2})]\!]_\Phi \approx [\![(\{N_1\}_{L_1'},\{N_2\}_{L_2'})]\!]_\Phi$ implies that the corresponding lengths in the two encryption terms must be the same: $l(\{M_1\}_{L_1}) = l(\{N_1\}_{L_1'})$ and $l(\{M_2\}_{L_2}) = l(\{N_2\}_{L_2'})$, which then implies that $(\Box_{l(\{M_1\}_{L_1})},\Box_{l(\{M_2\}_{L_2})}) = (\Box_{l(\{N_1\}_{L_1'})},\Box_{l(\{N_2\}_{L_2'})})$. Therefore, $(\{M_1\}_{L_1},\{M_2\}_{L_2}) \cong (\{N_1\}_{L_1'},\{N_2\}_{L_2'})$. In conclusion, the formal logic we introduced for our implementation of the OTP is complete.

Corollary 77 (OTP Completeness). *Let M and N be two valid expressions in* $\boldsymbol{Exp}_{\mathrm{OTP}}$. *If $[\![M]\!]_\Phi$ and $[\![N]\!]_\Phi$ have the same probability distributions, then $M \cong_{\mathit{OTP}} N$.*

6 Conclusions and Further Work

We have studied extensions of the Abadi-Rogaway logic of indistinguishability for formal cryptographic expressions, considering and solving two problems that were left uncovered by the original result.

The first uncovered problem is the case of soundness in the presence of key cycles. Computational soundness for expressions *without* key cycles was proved in Abadi and Rogaway [3] under the assumption that a computational encryption scheme satisfies a strong version of semantic security (type-0). We have considered a modification of their logic in the case of encryption schemes both which-key revealing and message-length revealing. In the presence of key cycles, we have proved that the computational soundness property follows from the key-dependent message (KDM) security proposed by Black *et al.* [19]. We obtain our soundness result by strengthening the computational model rather than weakening the formal model. We have also shown that the computational soundness property neither implies nor is implied by type-0 security, and thus the original Abadi-Rogaway result could not have been demonstrated for key cycles using the security notions described in their work. We refer the reader to [4] for a discussion of soundness in the presence of key cycles for the case of public-key encryption. Similarly to the symmetric-key setting, it is shown that soundness follows from (public-) KDM security, and that soundness does not imply, nor is implied by, CCA-2 security.

The other uncovered problem of the original Abadi-Rogaway result addressed in this paper concerned the possibility of leakage of information by an encryption scheme. As said before, the original result assumed a very strong notion of security (type-0) which is not actually achieved by many encryption schemes. Thus, one might wonder if a similar result might be derived for weaker schemes. We have showed that for symmetric encryption, subtle differences between security definitions can be faithfully reflected in the formal symbolic setting. We have introduced a general probabilistic framework which includes both the computational and the information-theoretic encryption schemes as special cases. We have established soundness and completeness theorems in this general framework, as well as new applications to specific settings: an information-theoretic interpretation of formal expressions in One-Time Pad, and also computational interpretations in type-1 (length-revealing), type-2 (which-key revealing) and type-3 (which-key and length revealing) encryption schemes based on computational complexity.

Our work presents several directions for future research. Independently of any soundness considerations, several questions about KDM security remain unanswered. There is no known implementation of KDM security in the standard model, although there are several natural candidates (*e.g.,* Cramer-Shoup [25]). Conversely, there remains to be found a natural (*i.e.*, non-constructed) example of an encryption scheme which is secure in the sense of type-0 (or CCA-2) but is not KDM-secure. Further, even the constructed examples fail to provide KDM security only when presented with key cycles of length 1. It may in fact be possible that type-0/CCA-2 security implies KDM security when all key cycles are of length 2 or more.

With regard to soundness in the presence of key cycles, it seems desirable to extend our results from the passive-adversary setting to that of the active adversary. Also, our results do not completely unite the two models. We show that the relationship between the formal and computational models requires more than type-0/CCA-2 security. While it demonstrates that KDM security is also necessary, it does not show it to be sufficient—even when conjoined with CCA-2 security (asymmetric encryption). That is, this investigation is not complete; it is more than likely that additional properties will be revealed as soundness is more fully explored.

Also, one might consider various expansions of the formal setting that would allow additional operations such as *xor*, pseudorandom permutations, or exponentiation. Soundness and completeness of such richer formal settings would, of course, need exploration. In particular, the definition of patterns appears to be rather subtle in such richer settings. We would also like to understand how our methods fit with the methods of [42].

Lastly, one might consider exploring partial leakage in the setting of asymmetric encryption. One might also extend our methods and investigate formal treatment of other cryptographic primitives. It would be interesting to see if our methods could be combined with the methods of [11, 22].

Acknowledgments. We want to thank M. Abadi, M. Backes, J. Black, R. Canetti, A. Gordon, J. Guttman, S. Hohenberger, R. Küsters, A. Lysyanskaya, D. Micciancio, J. Mitchell, T. Shrimpton, D. Unruh, and B. Warinschi for their valuable comments and informative discussions. Some of our joint work was done during the Protocol eXchange meetings; we thank S. Pinsky, E. Zieglar, and G. Dinolt for organizing the meetings and providing a conducive and encouraging atmosphere. Lastly we thank the anonymous referees for their thoughtful comments and suggestions.

This work was done while the first author was a visiting student at the University of Pennsylvania.

References

1. M. Abadi and J. Jürjens. Formal eavesdropping and its computational interpretation. In N. Kobayashi and B. C. Pierce, editors, *Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Software (TACS)*, volume 2215 of *Lecture Notes in Computer Science*, pages 82–94, Sendai, Japan, October 29–31 2001. Springer.
2. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In J. van Leeuwen, O. Watanabe, M. Hagiya, P. D. Mosses, and T. Ito, editors, *Proceedings of the 1st IFIP International Conference on Theoretical Computer Science (IFIP TCS)*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22, Sendai, Japan, August 17–19 2000. Springer.
3. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, January 2002. Preliminary version presented at IFIP TCS'00.
4. P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In S. De Capitani di Vimercati, P. Syverson, and D. Gollmann, editors, *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *Lecture Notes in Computer Science*, pages 374–396, Milan, Italy, September 12–14 2005. Springer.

5. P. Adão, G. Bana, and A. Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW)*, pages 170–184, Aix-en-Provence, France, June 20–22 2005. IEEE Computer Society Press.
6. M. Backes and C. Jacobi II. Cryptographically sound and machine-assisted verification of security protocols. In H. Alt and M. Habib, editors, *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of *Lecture Notes in Computer Science*, pages 675–686, Berlin, Germany, February 27–March 1 2003. Springer.
7. M. Backes and B. Pfitzmann. A cryptographically sound security proof of the Needham-Schröeder-Lowe public-key protocol. *IEEE Journal on Selected Areas in Communications*, 22(10):2075–2086, December 2004. Preliminary version presented at FSTTCS'03.
8. M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable Dolev-Yao style cryptographic library. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW)*, pages 204–218, Pacific Grove, CA, USA, June 28–30 2004. IEEE Computer Society Press. Full version available at IACR ePrint Archive, Report 2004/059.
9. M. Backes and B. Pfitzmann. Relating symbolic and cryptographic secrecy. *IEEE Transactions on Dependable and Secure Computing*, 2(2):109–123, April 2005. Preliminary version presented at S&P'05.
10. M. Backes, B. Pfitzmann, and M. Waidner. Secure asynchronous reactive systems. Available at IACR ePrint Archive, Report 2004/082.
11. M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, pages 220–230, Washington D.C., USA, October 27–30 2003. ACM Press. Full version available at IACR ePrint Archive, Report 2003/015, January 2003.
12. M. Backes, B. Pfitzmann, and M. Waidner. Symmetric authentication within a simulatable cryptographic library. *International Journal of Information Security*, 4(3):135–154, June 2005. Preliminary version presented at ESORICS'03.
13. Michael Backes, Birgit Pfitzmann, and Michael Waidner. A composable cryptographic library with nested operations (extended abstract). In S. Jajodia, V. Atluri, and T. Jaeger, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, Washington D.C., USA, October 27–30 2003. ACM Press. Preprint on IACR ePrint 2003/015.
14. G. Bana. *Soundness and Completeness of Formal Logics of Symmetric Encryption*. PhD thesis, University of Pennsylvania, July 2004. Available at IACR ePrint Archive, Report 2005/101, April 2005.
15. M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the The 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 652–663, Lisbon, Portugal, July 11–15 2005. Springer.
16. D. Beaver. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2):75–122, 1991.
17. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, pages 394–403, October 1997.
18. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–

45, Santa Barbara, CA, USA, August 23–27 1998. Springer. Full version available at `http://www.cs.ucsd.edu/users/mihir/papers/relations.html`.

19. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In K. Nyberg and H. Heys, editors, *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography (SAC)*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75, St. John's, Newfoundland, Canada, August 15–16 2002. Springer.
20. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 98–118, Innsbruck, Austria, May 6–10 2001. Springer.
21. R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
22. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–145, Las Vegas, NV, USA, October 14–17 2001. IEEE Computer Society Press. Full version available at IACR ePrint Archive, Report 2000/067.
23. R. Canetti and J. Herzog. Universally composable symbolic analysis of mutual authentication and key-exchange protocols, March 5–7 2006. To Appear. Full version available at IACR ePrint Archive, Report 2004/334.
24. V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In M. Sagiv, editor, *Proceedings of the 14th European Symposium on Programming (ESOP)*, volume 3444 of *Lecture Notes in Computer Science*, pages 157–171, Edinburgh, UK, April 4–8 2005. Springer.
25. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Advances in Cryptology CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Santa Barbara, CA, USA, August 23–27 1998. Springer.
26. A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the The 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 16–29, Lisbon, Portugal, July 11–15 2005. Springer.
27. A. Datta, R. Küsters, J. C. Mitchell, and A. Ramanathan. On the relationships between notions of simulation-based security. In J. Kilian, editor, *Proceedings of the 2nd Theory of Cryptography Conference (TCC)*, volume 3378 of *Lecture Notes in Computer Science*, pages 476–494, Cambridge, MA, USA, February 10–12 2005. Springer.
28. D. Dolev and A. C. Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, March 1983. Preliminary version presented at FOCS'81.
29. S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 77–93, Santa Barbara, CA, USA, August 11–15 1991. Springer.
30. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and Systems Sciences*, 28(2):270–299, April 1984. Preliminary version presented at STOC'82.
31. J. D. Guttman, F. J. Thayer, and L. D. Zuck. The faithfulness of abstract protocol analysis: Message authentication. In P. Samarati, editor, *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS)*, pages 186–195, Philadelphia, PA, USA, November 05–08 2001. ACM Press.
32. J. Herzog. Computational Soundness of Formal Adversaries. Master thesis, Massachussets Institute of Technology, 2002.

33. J. Herzog. *Computational Soundness for Standard Assumptions of Formal Cryptography*. PhD thesis, Massachussets Institute of Technology, May 2004. Available at `http://theory.lcs.mit.edu/∼jherzog/papers/herzog-phd.pdf`.
34. J. Herzog, M. Liskov, and S. Micali. Plaintext awareness via key registration. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 548–564, Santa Barbara, CA, USA, August 17–21 2003. Springer.
35. O. Horvitz and V. Gligor. Weak key authenticity and the computational completeness of formal encryption. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 530–547, Santa Barbara, CA, USA, August 17–21 2003. Springer.
36. R. Impagliazzo and B. M. Kapron. Logics for reasoning about cryptographic constructions. In *44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 372–383, Cambridge, MA, USA, October 11–14 2003. IEEE Computer Society Press.
37. R. Janvier, Y. Lakhnech, and L. Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In M. Sagiv, editor, *Proceedings of the 14th European Symposium on Programming (ESOP)*, volume 3444 of *Lecture Notes in Computer Science*, pages 172–185, Edinburgh, UK, April 4–8 2005. Springer.
38. P. Laud. Encryption cycles and two views of cryptography. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC)*, number 31 in Karlstad University Studies, pages 85–100, Karlstad, Sweden, November 7–8 2002.
39. P. Laud. Symmetric encryption in automatic analyses for confidentiality against active adversaries. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P)*, pages 71–85, Oakland, CA, USA, May 9–12 2004. IEEE Computer Society Press.
40. P. Laud and R. Corin. Sound computational interpretation of formal encryption with composed keys. In J. I. Lim and D. H. Lee, editors, *Proceedings of the 6th International Conference on Information Security and Cryptology (ICISC)*, volume 2971 of *Lecture Notes in Computer Science*, pages 55–66, Seoul, Korea, November 27–28 2003. Springer.
41. P. Lincoln, J. C. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic polynomial-time framework for protocol analysis. In M. Reiter, editor, *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS)*, pages 112–121, San Francisco, CA, USA, November 3–5 1998. ACM Press.
42. U. Maurer. Indistinguishability of random systems. In L. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 110–132, Amsterdam, The Netherlands, April 28–May 2 2002. Springer.
43. S. Micali and P. Rogaway. Secure computation. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 392–404, Santa Barbara, CA, USA, August 11–15 1991. Springer.
44. D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In J. Kilian, editor, *Proceedings of the 2nd Theory of Cryptography Conference (TCC)*, volume 3378 of *Lecture Notes in Computer Science*, pages 169–187, Cambridge, MA, USA, February 10–12 2005. Springer.
45. D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *Journal of Computer Security*, 12(1):99–130, 2004. Preliminary version presented at WITS'02.
46. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In M. Naor, editor, *Proceedings of the 1st Theory of Cryptography Conference (TCC)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151, Cambridge, MA, USA, February 19–21 2004. Springer.
47. J. C. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague. A probabilistic polynomial-time calculus for the analysis of cryptographic protocols. Full, revised version available

at http://theory.stanford.edu/people/jcm/publications.htm. Preliminary version under the title "Probabilistic Bisimulation and Equivalence for Security Analysis of Network Protocols" in FOSSACS'04, Springer LNCS Vol. 2987.

48. B. Pfitzmann, M. Schunter, and M. Waidner. Cryptographic security of reactive systems. *Electronic Notes in Theoretical Computer Science*, 32:59–77, 2000. Preliminary version presented at WSAIF'99.
49. B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In P. Samarati, editor, *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS)*, pages 245–254, Athens, Greece, November 01–04 2000. ACM Press. Extended version (with M. Shunter) available as IBM Research Report RZ 3206, 2000, http://www.zurich.ibm.com/security/models.
50. B. Pfitzmann and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P)*, pages 184–200, Oakland, CA, USA, May 14–16 2001. IEEE Computer Society Press.
51. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Santa Barbara, CA, USA, August 11–15 1991. Springer.
52. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 543–553, New York, NY, USA, October 17–19 1999. IEEE Computer Society Press.
53. B. Warinschi. A computational analysis of the Needham-Schröeder-(Lowe) protocol. In *Proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW)*, pages 248–262, Pacific Grove, CA, USA, June 30–July 2 2003. IEEE Computer Society Press.
54. A. C. Yao. Theory and applications of trapdoor functions. In *23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, Chicago, IL, USA, November 3–5 1982. IEEE Computer Society Press.