

Soundness of Formal Encryption in the Presence of Key-Cycles

Pedro Adão¹, Gergei Bana², Jonathan Herzog³, and Andre Scedrov²

¹ Center for Logic and Computation, IST, Lisboa, Portugal

² Department of Mathematics, University of Pennsylvania, Philadelphia, USA

³ The MITRE Corporation

pad@math.ist.utl.pt {bana, scedrov}@math.upenn.edu jherzog@mitre.org

Abstract. Both the formal and computational models of cryptography contain the notion of message *equivalence* or *indistinguishability*. An encryption scheme provides *soundness* for indistinguishability if, when used to map formal messages into the computational model, equivalent formal messages become indistinguishable computational distributions. Previous soundness results are limited in that they do not apply when *key cycles* are present. We demonstrate that an encryption scheme provides soundness in the presence of key cycles if it also satisfies the recently-introduced notion of *key-dependent message* (KDM) security. We also show that soundness in the presence of key-cycles (and KDM security) neither implies nor is implied by security against chosen ciphertext attack (CCA-2). Therefore, soundness for key-cycles is possible using new definitions, not possible using previous definitions, and there is more to relating the formal and computational models than chosen-ciphertext security.

1 Introduction

‘Security’ is the Rorschach blob of theoretical computer science: every model of computation has attempted to define it in its own way. In the area of cryptographic protocols, two models are noteworthy for their natural definitions and rigorous proofs. The first of these models, the *computational model*, is derived from 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*, is named such 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, we consider the relationship between these two models. There are two key differences between them: their representations of messages and the powers they give to the adversary.

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

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. In 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 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. Formal equivalence maps directly onto computational indistinguishability, as is demonstrated via a natural computational *interpretation* of expressions. This function, parameterized by a computational encryption scheme, 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 if all pairs of equivalent formal messages map to indistinguishable probability distribution ensembles. If so, we say that *soundness* holds⁴ and (intuitively) it implies that the formal model is a faithful abstraction of the computational model in this regard.

The first soundness result of this type is due to Abadi and Rogaway in the symmetric-key encryption setting [1]. They demonstrated that soundness holds when the security level of the computational encryption algorithm is ‘type-0,’ a property of their own devising. This result was later translated to the public key setting (which is also the setting we will consider in this paper) by Micciancio and Warinschi [21]. They found that soundness in this setting is guaranteed by encryption schemes that satisfy the standard definition of chosen-ciphertext security (CCA-2 in the notation of [6]). This power of chosen-ciphertext security has been confirmed by subsequent extensions [11, 9]. However, both the original result of Abadi and Rogaway and the later extensions (including those that use CCA-2 security) share a common limitation: they do not necessarily apply in the presence of key-cycles.

A persistent question. A formal message M contains a *key-cycle* if it contains encryption terms $\{M_1\}_{K_1}, \{M_2\}_{K_2}, \dots, \{M_n\}_{K_n}$ (where $\{M_i\}_{K_i}$ denotes the encryption of the message M_i with the public key K_i) such that the M_i contains the key necessary to decrypt $\{M_{i+1}\}_{K_{i+1}}$ and M_n contains the key necessary to decrypt $\{M_1\}_{K_1}$. The simplest key-cycle is the message $\{K^{-1}\}_K$, where K^{-1} denotes the (private) decryption key associated with the encryption key K , but more complex key cycles are possible (e.g., $\{K_2^{-1}\}_{K_1} \{K_1^{-1}\}_{K_2}$).

The formal model makes no distinction between key cycles and any other sequence of encryptions. Likewise, the presence of a key cycle will not prevent a formal expression from being interpreted as a computational distribution ensemble in the natural way. However, neither the soundness result of Abadi and Rogaway nor subsequent soundness demonstrations (described in Section 2) are known to hold for such messages. (In fact, the stronger of these results [4, 9] assume that no private or symmetric keys are encrypted at all!)

Thus, the question of key-cycles is both interesting in its own right and has implications in a larger context. The standard security definitions for computational encryption, such as CCA-2 security, do not obviously imply security in the presence of key-cycles [18]. The formal model, on the other hand, assumes that key-cycles do not weaken encryption in any way. Therefore, the somewhat esoteric-seeming issue of key-cycles may represent a “gap” between the formal and computational models, and shed light on their general relationship.

Gaps between the two models. Because these two models came from very different fields, it is somewhat surprising that they can be related in any way at all. However, the majority of the results show the formal model to be sound with respect to standard definitions of the computational model—with some notable exceptions. Some aspects of the formal model have been shown to be overly strong relative to the computational model. The original soundness results of Abadi and Rogaway, for example, assumed that formal encryption concealed all aspects of the plaintext. In particular, their result requires that symmetric encryption hide the length of the plaintext (among other things). Unfortunately, this cannot be achieved for many contexts. Soundness in these other contexts is considered by Micciancio and Warinschi [21], Bana [5], Micciancio and Panjwani [19] and Adão, Bana and Scedrov [2], who require a weaker notion of formal

⁴ 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.

equivalence. (In keeping with this, we will use the unfortunately more complex formal model that addresses these weaknesses.)

On the other hand, other aspects of the formal model have been shown to be overly weak compared to the computational model. Canetti and Herzog [9], for example, have demonstrated that the formal definition of secrecy (in the context of key-exchange protocols) is strictly weaker than the computational definition. That is, some protocols may satisfy the formal notion of security but not the computational one. Having demonstrated this gap, the authors close it by providing a strictly stronger formal definition that does abstract the computational definition in a demonstrably faithful way.

Thus, at least two “gaps” between the formal and computational models have been uncovered. In both cases, their resolution forced changes to the formal model. Thus, the question of key cycles raises larger issues. Do key-cycles represent an actual “gap” between the two models? If so, will its resolution again cause changes to the formal model, or could it be resolved through modifications to the computational model this time?

An alternate approach. Laud [15] has proposed a solution to the problem of key cycles which takes the first approach. That is, Laud’s proposed 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 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 actual extent to which the computational adversary can distinguish messages. It is often unreasonable from a cryptographer’s point of view to assume that the computational adversary can always break 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 but by strengthening it in the computational one.

Our work. In this paper, we resolve the issue of soundness in the presence of key-cycles by using the notion of *key-dependent message* (KDM) security for asymmetric encryption. This definition was recently introduced both by Black, Rogaway and Shrimpton [7], who consider it in their own right, and simultaneously by Camenisch and Lysyanskaya [8], 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. That is, a KDM-secure encryption scheme provides soundness for the existing and unweakened formal model.
2. More surprisingly, soundness *requires* new computational definitions of security. That is, we demonstrate that soundness and KDM security neither imply nor are implied by chosen-ciphertext (CCA-2) security, the strongest known definition of security in the (standard) computational model.⁵

Thus, the problem of key cycles was a genuine gap between the formal and computational models at the time of the original Abadi-Rogaway result, but is no longer due to 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 (in the case of asymmetric encryption) than chosen-ciphertext security.

⁵ A stronger notion of security, plaintext-awareness, is known, but it is defined (generally) only in the random-oracle model and so is regarded as non-standard. See Herzog, Liskov and Micali [12] for fuller discussion and an alternate definition.

Limitations. We note that our results contain a few limitations of their own. Firstly, we consider a passive adversary only. Secondly, KDM security has only been actually implemented in the random oracle model, a non-standard variation of the computational model. Lastly, we use a weakened version of the formal model in which encryptions reveal the length of the plaintext and the key used to encrypt. (Rephrased, we consider ‘type-3’ encryption and not ‘type-0.’)

However, it should also be noted that these limitations are smaller than they may first seem. We consider a passive adversary solely for simplicity. We expect that our results can be extended to consider active adversaries (as was the original Abadi-Rogaway result) and regard our work as a ‘first step’ towards that extension. Secondly, we do not use the random oracle in this work. We use only the definition of KDM security, which is well-founded in the standard computational model, and so do not rely upon the random oracle. Lastly, the issue of type-3 vs. type-0 encryption is orthogonal to our work. We express our definitions and results in the style of type-3 encryption for two reasons: to be in keeping with recent extensions, and only type-3 security is guaranteed by the standard computational definitions. (That is, definitions such as chosen-ciphertext security do not *a priori* conceal the encryption key or the length of the plaintext.) However, our results will map directly to their type-0 analogies provided that the computational encryption scheme is length- and key-concealing as well as being KDM-secure.

Overview of the paper. We begin with a discussion of some previous work (Section 2). We then present (Section 3) modified versions of Abadi and Rogaway’s soundness definition and result. As mentioned above, we consider encryption schemes that reveal the key used to encrypt and the length of the plaintext.

We then show that (adaptive) chosen-ciphertext security alone cannot ensure soundness in the presence of key-cycles (Section 4). Thus, soundness for key cycles could not have been demonstrated with the computational definitions available to Abadi and Rogaway, and new definitions were necessary.

We then present the notion of KDM security (Section 5.1) and show that it is strong enough to imply soundness in the presence of key cycles (Section 5.2). We also show (Section 5.3) that KDM-security is in fact a new notion: it neither implies nor is implied by CCA-2 security. To finish our discussion on the relationships between the different security notions, we show that soundness does not imply IND-CPA security.

We conclude (Section 6) with the discussion of some future work.

Acknowledgements. We want to thank J. Black, A. Gordon, S. Hohenberger, A. Lysyanskaya and T. Shrimpton for their valuable comments and informative discussions.

2 Previous Work

The effort to connect these two models—formal and computational—has attracted researchers in the last few years. Examples of this are the works of Lincoln, Mitchell, Mitchell and Scedrov [?], where they developed a framework for analyzing security protocols based on a probabilistic process algebra (this work was later extended by Mitchell, Ramanathan, Scedrov and Teague [?]) and Guttman, Thayer and Zuck [?], where the authors try to quantify how much a concrete implementation of a primitive may diverge from the abstract primitive.

Canetti [?] proposes a general framework that guarantees security even when a secure protocol is used as a component of an arbitrary system. He formulates universally composable definitions for cryptographic properties such as authentication, key-exchange, zero-knowledge, among others. This work was later extended by Canetti, Lindell, Ostrovsky and Sahai [?] where a secure realization of any multi-party functionality in a universally composable way is given, regardless the number of corrupted participants.

Pfitzmann, Schunter and Waidner [?], discuss relations between cryptographic and abstract definitions of security and introduce a composition theorem for cryptographic protocols. Backes, Jacobi and Pfitzmann [?]

and Backes, Pfitzmann and Waidner [4] extend the previous work by proving soundness for general trace-based properties in the presence of active adversaries.

Independently, Backes and Pfitzmann [3] and Canetti and Herzog [9] present a new definition of (formal) secrecy for key-exchange that is sound with respect to a real implementation of a key-exchange protocol in the reactive-simulatability scenario, for the former, and in a universally composable scenario for the latter.

Two cryptographically sound security proofs for the Needham-Schroeder-Lowe protocol have been given independently. Backes and Pfitzmann [?] prove that NSL protocol is secure in the presence of active attacks using their abstract cryptographic library (that is a secure implementation of a real cryptographic library) while Warinschi [24] shows that the NSL protocol is a secure mutual authentication protocol when the encryption scheme satisfies indistinguishability under chosen-ciphertext attack. More, he shows that Lowe's attack to the original protocol can naturally be cast to the computational framework and proves that chosen-plaintext security for encryption schemes is not sufficient to ensure soundness of formal proofs with respect to the computational setting.

Another way of trying to bridge the gap between the two models, the one that we follow in this paper, was proposed by Abadi and Rogaway [1]. They show that sufficiently strong cryptography enforced computational soundness for a notion of formal equivalence. From this, many other results followed: Abadi and Jürjens [?] extend this soundness result from a single message to messages being exchanged over time. Bana [5] and Adão *et al.* [2] extend the original Abadi-Rogaway result to weaker encryption schemes, while Laud and Corin [17] do the same for composite keys. These two extensions are orthogonal: the former extends the applicability of the result to other encryption schemes (e.g., encryption schemes that reveal the length of the underlying plaintext) while the latter extends the set of expressions of the symbolic model. Herzog *et al.* [12] demonstrate soundness for non-malleability properties, and Herzog [11] later shows that this soundness for non-malleability is in fact implied by soundness of indistinguishability.

The above results consider only passive adversaries. Active adversaries were considered in the works of Backes *et al.* and Canetti and Herzog mentioned above, as well as in the work of Micciancio and Warinschi [21] that independently demonstrate soundness (in a public-key setting) for mutual-authentication properties in the presence of active adversaries. All these results do not allow the use of secret keys as messages and ciphertext forwarding.

Recent extensions of [21] by Micciancio and Panjwani [19] prove soundness of a group-key distribution protocol in the presence of a CPA-secure scheme. Nevertheless, in spite of the requirement for a weaker encryption scheme, they have the same two syntactic restrictions of [21]. Cortier and Warinschi [10] also extend [21]. They explore the use of automated tools to prove security (trace) properties. They prove that for the case of protocols that use nonces, signatures, asymmetric encryption and allow ciphertext forwarding, symbolic integrity and secrecy proofs are sound with respect to the computational model.

Janvier, Lakhnech and Mazaré [14] also extend the syntax of the logic to allow signature schemes. They show that even in the presence of asymmetric encryption (that is IND-CCA secure) and digital signatures (that are UNF-CCA secure) the formal model constitutes a safe abstraction of the computational model. They also allow the encryption of secret keys as messages and message forwarding. Another result involving active adversaries is Laud [16]. He proves soundness of confidentiality properties for symmetric encryption in a model with a fixed number of sessions.

These results however, do not hold in the presence of key-cycles. As we mentioned in the introduction this problem was addressed by Laud [15] in a different way from the one that we will address in this paper.

The connection between these two models is not one-way, that is, there is also research regarding the other direction, *completeness*. (That is, an interpretation enforces completeness if two formal messages must be equivalent whenever their interpretations are indistinguishable.) Micciancio and Warinschi [20] show that a sufficiently strong encryption scheme enforces completeness for indistinguishability properties. This result was strengthened by Horvitz and Gligor [13] where an exact characterization of the computational requirements on the encryption scheme under which completeness holds is given. Later, it was shown by

Bana [5] and Adão *et al.* [2] that completeness also holds for a more general class of (weaker) encryption systems. We do not discuss completeness further in this work, for the issue of key-cycles does not arise when dealing with this property.

3 Computational Soundness for Indistinguishability

We start presenting the formal model, and then describe the computational model in a fairly standard way. Lastly, we introduce the notion of soundness we consider in this paper: that equivalent formal expressions represent indistinguishable computational distribution ensembles.

In general, this is almost entirely a modified version of the treatment of Abadi and Rogaway [1]. The differences are that we deal with asymmetric encryption, formal encryptions reveal the keys used to encrypt, and formal expressions have an associated ‘length.’

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 (Expressions). Let $\mathbf{Keys} = \{K_1, K_2, K_3, \dots\}$ be an infinite discrete set of symbols, called the set of encryption keys, and $\mathbf{Keys}^{-1} = \{K_1^{-1}, K_2^{-1}, K_3^{-1}, \dots\}$ the corresponding set of decryption keys. Let $\mathbf{Blocks} = \{0, 1\}^*$, the set of finite bit-strings. We define the set of expressions, \mathbf{Exp} , by the grammar:

$$\mathbf{Exp} ::= \mathbf{Keys} \mid \mathbf{Keys}^{-1} \mid \mathbf{Blocks} \mid (\mathbf{Exp}, \mathbf{Exp}) \mid \{\mathbf{Exp}\}_{\mathbf{Keys}}$$

We will denote by $\mathbf{Keys}(M)$ the set of all encryption keys occurring in M and by $\mathbf{Keys}^{-1}(M)$ the set of decryption keys in M . Expressions of the form $\{N\}_K$ 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 (such as the public keys and compromised private keys).

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. To rigorously define this notion we need to formalize when a ciphertext 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 can then use them to decrypt encryption terms of the expression. This decryption might reveal yet more keys. By repeating this process, the adversary can learn the set of *recoverable decryption keys*:

Definition 2 (Visible Subterms, Recoverable Decryption Keys). Let $\mathit{vis}(M) \subseteq \mathbf{Exp}$, the visible subterms of M , be the smallest set of expressions containing M such that:

1. $(N_1, N_2) \in \mathit{vis}(M) \implies N_1 \in \mathit{vis}(M)$ and $N_2 \in \mathit{vis}(M)$, and
2. $\{N\}_K$ and $K^{-1} \in \mathit{vis}(M) \implies N \in \mathit{vis}(M)$.

Let $R\text{-Keys}(M)$, the set of recoverable decryption keys in M , be $\mathit{vis}(M) \cap \mathbf{Keys}^{-1}$.

This allows us to identify those encryption terms of an expression that will be “opaque” to the adversary: those protected by at least one non-recoverable decryption key. Thus, we wish to say that two expressions are equivalent if they differ only in the contents of their “opaque” encryption terms.

However, computational realities force us to add two ways in which an opaque encryption may leak information: they now reveal the key used to encrypt, and they now reveal the ‘length’ of the plaintext. This second condition requires that the notion of length be added to the formal model [20, 11, 5]:

Definition 3 (Formal Length). We introduce a fresh letter ℓ which will work as a function symbol on expressions with the following identities:

- For all blocks B_1 and B_2 , $\ell(B_1) = \ell(B_2)$ iff $|B_1| = |B_2|$ (the number of symbols in B_1 and B_2),
- $\forall i, j \in \mathbb{N}$, $\ell(K_i) = \ell(K_j)$ and $\ell(K_i^{-1}) = \ell(K_j^{-1})$,
- If $\ell(M_1) = \ell(N_1)$, $\ell(M_2) = \ell(N_2)$ then $\ell((M_1, M_2)) = \ell((N_1, N_2))$, and
- If $\ell(M) = \ell(N)$, then for all K_i , $\ell(\{M\}_{K_i}) = \ell(\{N\}_{K_i})$.

We introduce this function in order to be able to talk about the length of a formal expression. We need this to express that the encryption may leak information about the length. We note, that had we chosen **Blocks** to be just $\{0, 1\}$, then $\ell(M) = \ell(N)$ would have meant exactly that the type-trees of M and N are identical.

Remark 1. The addition of lengths to the formal model is fairly recent, and is not necessary for soundness if computational encryption can hide the length of the plaintext.

Recall that our goal is to define the “observable” portion of an expression, also called its *pattern*, so that we can define two expressions to be equivalent when their observable portions are identical:

Definition 4 (Pattern). We define the set of patterns, *Pat*, by the grammar:

$$\mathbf{Pat} ::= \mathbf{Keys} \mid \mathbf{Keys}^{-1} \mid \mathbf{Blocks} \mid (\mathbf{Pat}, \mathbf{Pat}) \mid \{\mathbf{Pat}\}_{\mathbf{Keys}} \mid \square_{\mathbf{Keys}, \ell(\mathbf{Exp})}$$

For an expression M , the pattern of M , denoted by $\text{pattern}(M)$, is obtained from M by replacing each encryption term $\{M'\}_K \in \text{vis}(M)$, for which $K^{-1} \notin R\text{-Keys}(M)$, by $\square_{K, \ell(M')}$.

For two patterns M and N , $M = N$ is defined the following way:

- If $M \in \mathbf{Blocks} \cup \mathbf{Keys} \cup \mathbf{Keys}^{-1}$, then $M = N$ iff M and N are identical.
- If M is of the form $\square_{K, \ell(M')}$, then $M = N$ iff N is of the form $\square_{K, \ell(N')}$, and $\ell(M') = \ell(N')$ in the sense of Definition 3.
- If M is of the form (M_1, M_2) , then $M = N$ iff N is of the form (N_1, N_2) where $M_1 = M_2$ and $N_1 = N_2$.
- If M is of the form $\{M'\}_K$, then $M = N$ iff N is of the form $\{N'\}_K$ where $M' = N'$.

The symbol, $\square_{K, \ell(M')}$ in a pattern reveals that the expression M' was encrypted with the key K and its length is $\ell(M')$ (Abadi and Rogaway replace these undecryptable terms by \square).

One last complication remains before we can define formal equivalence. The expressions K_1 and K_2 are different formal expressions even though they both have the same meaning: a randomly drawn key. We wish to formalize the notion of equivalence in such a way that these differences can be disregarded. Therefore, two formal expressions should be equivalent if they differ only on the names given to keys:

Definition 5 (Key-Renaming Function). A bijection $\sigma : \mathbf{Keys} \rightarrow \mathbf{Keys}$ 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 $\sigma(K)$ (including those occurrences as indices of \square) and all occurrences of keys K^{-1} in M by $(\sigma(K))^{-1}$.

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

Definition 6 (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 $\text{pattern}(M) = \text{pattern}(N\sigma)$.

3.2 The Computational Model

The fundamental objects of the computational world are strings, strings = $\{0, 1\}^*$, and families of probability distributions over strings. These families are indexed by a *security parameter* $\eta \in \mathbb{N}$ (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* if no efficient algorithm can determine from which distribution a value was sampled.

Definition 7 (Negligible Function). A function $\text{neg}(\eta) : \mathbb{N} \rightarrow \mathbb{R}$ is said to be negligible, if for any $c > 0$, there is an $n_c \in \mathbb{N}$ such that $\text{neg}(\eta) \leq \eta^{-c}$ whenever $\eta \geq n_c$.

Definition 8 (Indistinguishability). Two distributions from $\{D_\eta\}_{\eta \in \mathbb{N}}$ and $\{D'_\eta\}_{\eta \in \mathbb{N}}$, are indistinguishable, written $D_\eta \approx D'_\eta$, if for all PPT adversaries A and for all sufficiently large η ,

$$|\Pr [d \leftarrow D_\eta; A(1^\eta, d) = 1] - \Pr [d \leftarrow D'_\eta; A(1^\eta, d) = 1]| \leq \text{neg}(\eta)$$

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

Definition 9 (Encryption Scheme). A computational asymmetric encryption scheme is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where:

- $\mathcal{K} : \text{parameters} \times \text{coins} \rightarrow \text{publickey} \times \text{secretkey}$ is a key-generation algorithm with $\text{parameters} = \mathbb{N}$,
- $\mathcal{E} : \text{publickey} \times \text{plaintexts} \times \text{coins} \rightarrow \text{ciphertexts}$ is an encryption function, and
- $\mathcal{D} : \text{secretkey} \times \text{strings} \rightarrow \text{plaintexts}$ is such that for all $(e, d) \in \text{publickey} \times \text{secretkey}$ and $\omega \in \text{coins}$

$$\mathcal{D}(d, \mathcal{E}(e, m, \omega)) = m \text{ for all } m \in \text{plaintexts}.$$

All these algorithms are computable in polynomial time in the size of the input. (For this reason, the set parameters is usually represented as 1^* .) We insist that $|\mathcal{E}(e, m, w)| = |\mathcal{E}(e, m, w')|$ for all $e \in \text{publickey}$, $m \in \text{plaintexts}$ and $w, w' \in \text{coins}$, where $|x|$ stands for the binary length of x . We also insist that $0^* \subseteq \text{plaintexts}$.

We also insist that for all e, x , all elements in the support of $\mathcal{E}(e, x)$ are of the same length and that this length depends only on $|x|$ and η (when $(e, d) \leftarrow \mathcal{K}(1^\eta)$). We also insist that the set $0^* \subseteq \text{plaintexts}$.

3.3 Relating the Two Models

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 strings. As in Abadi and Rogaway [1], this interpretation is defined in an algorithmic way. The full formalism is given in Appendix B, but intuitively for expressions:

- Blocks are interpreted as 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 encryption algorithm.

We will denote by $\llbracket M \rrbracket_{\Phi_\eta}$ the distribution of $\Phi_\eta(M)$ and by $\llbracket M \rrbracket_\Phi$ the ensemble of $\{\llbracket M \rrbracket_{\Phi_\eta}\}_{\eta \in \mathbb{N}}$. For the interpretation of patterns, everything is the same as for the interpretation of expressions, but we also have:

- The interpretation of $\square_{K, \ell(M)}$ for a given security parameter η is the random variable belonging to the η in the interpretation of the expression $\{0^{|\Phi_\eta(M)|}\}_K$ where $|\Phi_\eta(M)|$ is the binary length of $\Phi_\eta(M)$ which is the same for all samples according to our assumptions about the encryption schemes. We can call the sequence $\{|\Phi_\eta(M)|\}_{\eta \in \mathbb{N}}$ the *interpretation* of $\ell(M)$.

For any pattern M , let $\Phi(M) = \{\Phi_\eta(M)\}_{\eta \in \mathbb{N}}$ be the family of random variables given by the interpretation, $\llbracket M \rrbracket_{\Phi_\eta}$ the distribution of $\Phi_\eta(M)$ and $\llbracket M \rrbracket_\Phi$ the ensemble of distributions $\{\llbracket M \rrbracket_{\Phi_\eta}\}_{\eta \in \mathbb{N}}$.

We can now define the notion of soundness.

Definition 10 (Soundness). *We say that an encryption scheme is sound, or provides soundness, if when used in the interpretation Φ*

$$M \cong N \Rightarrow \llbracket M \rrbracket_\Phi \approx \llbracket N \rrbracket_\Phi$$

for any expressions M and N .

The primary result of Abadi and Rogaway given in [1] is that, in the symmetric case, soundness is guaranteed by sufficiently strong cryptography (called “type-0”) if the expressions M and N have no key cycles. Subsequent work [21] translates this result to the setting of asymmetric encryption, and derives that a similar soundness property (in the absence of key cycles) is guaranteed by chosen-ciphertext security. Subsequent work [11, 9] confirm that chosen-ciphertext security suffices for several extensions, so long as key-cycles are prohibited. In the next section, we show that (surprisingly) chosen-ciphertext does not suffice for even this basic notion of soundness in the presence of key-cycles.

4 Chosen-Ciphertext Security is Not Enough

In this section we show that the standard notions of security, at the time that the Abadi and Rogaway results were published, were not strong enough to ensure soundness in the case of key-cycles. That is, it is possible to construct encryption schemes that satisfy the standard notions of security (CCA-2 in particular) but fail to provide soundness in the presence of key-cycles.

Definition 11 (Adaptive Chosen Ciphertext Security). *A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides indistinguishability under the adaptive chosen-ciphertext attack if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); \\ m_0, m_1 \leftarrow \mathcal{A}^{\mathcal{D}_1(\cdot)}(1^\eta, e); i \leftarrow \{0, 1\}; \\ c \leftarrow \mathcal{E}(e, m_i); g \leftarrow \mathcal{A}^{\mathcal{D}_2(\cdot)}(1^\eta, e, c) : \\ b = g] \leq \frac{1}{2} + \text{neg}(\eta)$$

The oracle $\mathcal{D}_1(x)$ returns $\mathcal{D}(d, x)$, and $\mathcal{D}_2(x)$ returns $\mathcal{D}(d, x)$ if $x \neq c$ and returns \perp otherwise. The adversary is assumed to keep state between the two invocations. It is required that m_0 and m_1 be of the same length.

That is, an adversary should not be able to learn from a ciphertext whether it contains the plaintext m_0 or the plaintext m_1 , even if:

- the adversary knows the public key used to encrypt,
- the adversary can choose the messages m_0 and m_1 itself, so long as the messages have the same length, and

- the adversary can request and receive the decryption of any *other* ciphertext.

This definition has been shown to be strictly stronger than almost all other definitions, including semantic security [6]. (See Appendix B for these other definitions.) It does not, however, guarantee soundness. The reason is, that A does not have (obviously) access to the private keys, and therefore the messages that he submits to the oracles \mathcal{D}_1 and \mathcal{D}_2 cannot depend on those private keys. Thus key-dependent messages are not considered and so, not captured:

Theorem 1. *CCA-2 security does not imply soundness. That is, if there exists an encryption scheme secure against the chosen-ciphertext attack, then there exists another encryption scheme which is secure against the chosen-ciphertext attack but does not provide soundness.*

Proof. This is shown via a simple counter-example. Assuming that there exists an encryption scheme secure against the chosen-ciphertext attack, we will use it to construct another scheme which is also secure against the chosen-ciphertext attack. 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 one of the simplest possible formal expressions with a key-cycle: $\{K^{-1}\}_K$. Let N be the expression $\{K_1^{-1}\}_{K_2}$. Since these two expressions are equivalent, an encryption scheme that enforces soundness requires that the family of distributions

$$\{(e, d) \leftarrow \mathcal{K}(1^\eta); c \leftarrow \mathcal{E}(e, d) : c\}_{\eta \in \mathbb{N}}$$

be indistinguishable from the family of distributions

$$\{(e_1, d_1) \leftarrow \mathcal{K}(1^\eta); (e_2, d_2) \leftarrow \mathcal{K}(1^\eta); c \leftarrow \mathcal{E}(e_1, d_2) : c\}_{\eta \in \mathbb{N}}.$$

However, this is not implied by Definition 11. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a CCA-2 secure encryption scheme. We construct a second CCA-2 secure encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ as follows:

- Let $\mathcal{K}' = \mathcal{K}$,
- Let $\mathcal{D}' = \mathcal{D}$, and
- Let \mathcal{E}' be the following algorithm:
 - Receive input (e, m) , an encryption key and a message;
 - Test whether m is the decryption key associated with e . For many encryption schemes, key-pairs are recognizable as such via number-theoretic properties. Even when this is not the case, this test can be conducted via the sub-algorithm:
 - * Select a random plaintext r ;
 - * Let $c \leftarrow \mathcal{E}(e, r)$;
 - * Let $p \leftarrow \mathcal{D}(m, c)$;
 - * Test whether $p = r$.
 - If m is the decryption key associated with e , output m ;
 - Otherwise, compute $c' \leftarrow \mathcal{E}(e, m)$ and output c' .

The scheme Π' acts exactly like Π unless the encryption algorithm \mathcal{E}' is called on a pair (e, m) where m (when used as a decryption key) can decrypt a random value encrypted with e . However, if such a value for m is easy to guess by the adversary, or easy to compute for a randomly generated public key e , then the scheme Π could not be CCA-2 secure. Thus, the new scheme Π' must also be CCA-2 secure. However, it does not guarantee indistinguishability for the two distributions above. The first distribution will output decryption key while the second outputs a ciphertext, and these two distributions are easily distinguished by form alone. \square

Since CCA-2 security implies a number of other definitions [6] (see Appendix A) we can easily conclude that these other definitions also do not imply soundness:

Corollary 1. *Soundness is not implied by any of: NM-CCA-1 security, IND-CCA-1 security, NM-CPA security, or IND-CPA (semantic) security.*

Therefore, soundness with key cycles could not have been demonstrated with the standard⁶ computational notions of security available at the time. In the next section, we show that this soundness property can, however, be met with new computational definitions.

5 KDM Security and Soundness for Key-Cycles

5.1 KDM-Security

In the last section, we showed that the standard notions of security are not strong enough to enforce soundness in the presence of key-cycles. However, KDM security, which was introduced by Black *et al.* [7] and independently by Camenisch and Lysyanskaya [8], is strong enough to enforce soundness even in the case of key-cycles. (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 strengthens IND-CPA (semantic) security. In case of semantic security, an adversary cannot distinguish between two oracles where one oracle encrypts the messages which the adversary submits, and the other oracle encrypts strings of all zeros (of the same length as the submitted messages). KDM strengthens this by allowing more general submissions to the oracles. In particular, in KDM security the adversary can submit not only fixed messages, but also *functions* of the decryption keys.

More precisely, KDM security is defined in terms of oracles $\text{Real}_{\mathbf{d}}$ and $\text{Fake}_{\mathbf{d}}$, which work as follows:

- Suppose that for a fixed security parameter $\eta \in \mathbb{N}$, a family of keys is given: $\{(e_i, d_i) \leftarrow \mathcal{K}(1^\eta)\}_{i \in \mathbb{N}}$. The adversary can now query the oracles providing them with a pair (j, g) , where $j \in \mathbb{N}$ and $g : \text{secretkey}^\infty \rightarrow \{0, 1\}^*$ is a constant length, deterministic function and \mathbf{d} is defined as the sequence $\langle d_1, d_2, \dots \rangle$:
 - The oracle $\text{Real}_{\mathbf{d}}$ when receiving this input returns $c \leftarrow \mathcal{E}(e_j, g(\mathbf{d}))$;
 - The oracle $\text{Fake}_{\mathbf{d}}$ when receiving this same input returns $c \leftarrow \mathcal{E}(e_j, 0^{|g(\mathbf{d})|})$.

The challenge facing the adversary is to decide with which oracle he has interacted, $\text{Real}_{\mathbf{d}}$ or $\text{Fake}_{\mathbf{d}}$. Formally:

Definition 12 (KDM Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme. Let the two oracles $\text{Real}_{\mathbf{d}}$ and $\text{Fake}_{\mathbf{d}}$ work as defined above. We say that the encryption scheme is KDM-secure if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\left| \Pr [(e, \mathbf{d}) \leftarrow \mathcal{K}(1^\eta) : A^{\text{Real}_{\mathbf{d}}}(1^\eta, e) = 1] - \Pr [(e, \mathbf{d}) \leftarrow \mathcal{K}(1^\eta) : A^{\text{Fake}_{\mathbf{d}}}(1^\eta, e) = 1] \right| \leq \text{neg}(\eta)$$

Remark 2. We note that although all known implementations of KDM-security are in the random-oracle model, this definition is well-founded even in the absence of this oracle. We also note that this definition is phrased in terms of indistinguishability. One could also imagine analogous definitions phrased in terms of non-malleability (Appendix A.2), but an exploration of those are beyond the scope of the paper.

⁶ Although plaintext awareness is stronger than CCA-2 encryption, but is defined only in the random-oracle model.

5.2 Soundness for Key-Cycles

Below, we present our main soundness result: if an encryption scheme is KDM secure, it also satisfies soundness.

Theorem 2 (KDM Security Implies Soundness). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a computational encryption scheme, Φ the interpretation of **Exp** for Π . If Π is KDM-secure, then Π satisfies soundness.*

This theorem holds even when the expressions have encryption-cycles. The proof in this case is a somewhat reduced hybrid argument. In a standard hybrid argument, like the one 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 $\llbracket M \rrbracket_\Phi$ is indistinguishable from $\llbracket \text{pattern}(M) \rrbracket_\Phi$. Then, since that holds for N too, and since $\text{pattern}(M)$ differs from $\text{pattern}(N)$ only in the name of keys, $\llbracket \text{pattern}(M) \rrbracket_\Phi$ is indistinguishable from $\llbracket \text{pattern}(N) \rrbracket_\Phi$, therefore the result follows. KDM security is used when we show that $\llbracket M \rrbracket_\Phi$ and $\llbracket \text{pattern}(M) \rrbracket_\Phi$ are indistinguishable.

Proof. For an arbitrary key K , let $\iota(K)$ denote the index of K . For an expression M , a set of formal decryption keys S , and a function τ defined on $(\mathbf{Keys} \cup \mathbf{Keys}^{-1}) \setminus S$ such that $\tau|_{\mathbf{Keys}}$ takes values in `publickey` and $\tau|_{\mathbf{Keys}^{-1}}$ takes values in `secretkey`, we define a function $f_{M,S,\tau} : \text{coins}^{e(M)} \times \text{secretkey}^\infty \rightarrow \text{strings}$ (where $e(M)$ is the number of encryptions in M) inductively the following way:

- For $B \in \mathbf{Blocks}$, let $f_{B,S,\tau} : \text{secretkey}^\infty \rightarrow \text{strings}$, $f_{B,S,\tau}(\mathbf{d}) = B$;
- For $K \in \mathbf{Keys}$, let $f_{K,S,\tau} : \text{secretkey}^\infty \rightarrow \text{strings}$, $f_{K,S,\tau}(\mathbf{d}) = \tau(K)$;
- For $K^{-1} \in \mathbf{Keys}^{-1}$, if $K^{-1} \notin S$, then $f_{K^{-1},S,\tau} : \text{secretkey}^\infty \rightarrow \text{strings}$, $f_{K^{-1},S,\tau}(\mathbf{d}) = \tau(K^{-1})$;
- For $K^{-1} \in \mathbf{Keys}$, if $K^{-1} \in S$, then $f_{K^{-1},S,\tau} : \text{secretkey}^\infty \rightarrow \text{strings}$, $f_{K^{-1},S,\tau}(\mathbf{d}) = d_{\iota(K)}$;
- Let $f_{(M,N),S,\tau} : \text{coins}^{e(M)} \times \text{coins}^{e(N)} \times \text{secretkey}^\infty \rightarrow \text{strings}$. Then, $f_{(M,N),S,\tau}$ is defined as $f_{(M,N),S,\tau}(\omega_M, \omega_N, \mathbf{d}) = [f_{M,S,\tau}(\omega_M, \mathbf{d}), f_{N,S,\tau}(\omega_N, \mathbf{d})]$;
- Let $f_{\{M\}_K,S,\tau} : \text{coins} \times \text{coins}^{e(M)} \times \text{secretkey}^\infty \rightarrow \text{strings}$. Then, $f_{\{M\}_K,S,\tau}$ is defined as $f_{\{M\}_K,S,\tau}(\omega, \omega_M, \mathbf{d}) = \mathcal{E}(\tau(K), f_{M,S,\tau}(\omega_M, \mathbf{d}), \omega)$.

The reader should note that the function $f_{M,S,\tau}$ defined above is constant length. This is due to the properties of the length function ℓ as well as the conditions in the definition of the encryption scheme.

We first prove that $\llbracket M \rrbracket_\Phi \approx \llbracket \text{pattern}(M) \rrbracket_\Phi$. Suppose that $\llbracket M \rrbracket_\Phi \not\approx \llbracket \text{pattern}(M) \rrbracket_\Phi$, which means that there is an adversary A that distinguishes the two distributions, that is

$$\Pr(x \leftarrow \llbracket M \rrbracket_{\Phi_\eta} : A(1^\eta, x) = 1) - \Pr(x \leftarrow \llbracket \text{pattern}(M) \rrbracket_{\Phi_\eta} : A(1^\eta, x) = 1)$$

is a non-negligible function of η . We will show that this contradicts the fact that the system is KDM-secure. To this end, we construct an adversary that can distinguish between the oracles Real_d and Fake_d . Let \mathcal{F} denote either of these oracles. Let $\mathbf{e} \in \text{publickey}^\infty$ be the array of public keys that \mathcal{F} outputs. From now on, let $S = \mathbf{Keys}^{-1} \setminus R\text{-Keys}(M)$, and if $K^{-1} \in S$, let then $\tau(K) = e_{\iota(K)}$. Consider now the following algorithm:

algorithm $B_\eta^{\mathcal{F}}(\mathbf{e}, M)$
 For $K^{-1} \in R\text{-Keys}(M)$, do $(\tau(K), \tau(K^{-1})) \leftarrow \mathcal{K}(1^\eta)$
 $y \leftarrow \text{CONVERT2}_{\mathbf{e}}(M, M)$
 $b \leftarrow A(1^\eta, y)$
return b

algorithm $\text{CONVERT2}_{\mathbf{e}}(M', M)$ with $M' \sqsubseteq M$

```

if  $M' = K$  where  $K \in \mathbf{Keys}$  then
  return  $\tau(K)$ 
if  $M' = K^{-1}$  where  $K^{-1} \in R\text{-Keys}(M)$  then
  return  $\tau(K^{-1})$ 
if  $M = B$  where  $B \in \mathbf{Blocks}$  then
  return  $B$ 
if  $M' = (M_1, M_2)$  then
   $x \leftarrow \text{CONVERT}_{2\mathbf{e}}(M_1, M)$ 
   $y \leftarrow \text{CONVERT}_{2\mathbf{e}}(M_2, M)$ 
  return  $[x, y]$ 
if  $M' = \{M''\}_K$  with  $K^{-1} \in R\text{-Keys}(M)$ , then
   $x \leftarrow \text{CONVERT}_{2\mathbf{e}}(M'', M)$ 
   $y \leftarrow \mathcal{E}(\tau(K), x)$ 
  return  $y$ 
if  $M' = \{M''\}_K$  with  $K^{-1} \notin R\text{-Keys}(M)$  and  $M'$  is not encrypted in  $M$  then
   $\omega \leftarrow \text{coins}^{\mathbf{e}(M')}$ 
   $y \leftarrow \mathcal{F}(\iota(K), f_{M'', S, \tau}(\omega, \cdot))$ 
  return  $y$ 

```

This algorithm applies the distinguisher $A(1^\eta, \cdot)$ on the distribution $\llbracket M \rrbracket_\Phi$ when \mathcal{F} is \mathbf{Real}_d , and the distribution of $\llbracket \text{pattern}(M) \rrbracket_\Phi$ when \mathcal{F} is \mathbf{Fake}_d . So, if $A(1^\eta, \cdot)$ can distinguish $\llbracket M \rrbracket_\Phi$ and $\llbracket \text{pattern}(M) \rrbracket_\Phi$, then $B_\eta^{\mathcal{F}}(\mathbf{e}, M)$ can distinguish \mathbf{Real}_d and \mathbf{Fake}_d . But we assumed that \mathbf{Real}_d and \mathbf{Fake}_d cannot be distinguished, so $\llbracket M \rrbracket_\Phi \approx \llbracket \text{pattern}(M) \rrbracket_\Phi$.

In a similar manner, we can show that $\llbracket N \rrbracket_\Phi \approx \llbracket \text{pattern}(N) \rrbracket_\Phi$. It is easy to see that $\llbracket \text{pattern}(M) \rrbracket_\Phi = \llbracket \text{pattern}(N) \rrbracket_\Phi$, because the two patterns differ only by key-renaming. Hence $\llbracket M \rrbracket_\Phi \approx \llbracket N \rrbracket_\Phi$. \square

Remark 3. Black *et al.* [7] realize KDM security in the random oracle model. However, our proof that KDM security implies soundness does not assume a particular model.

This one result has many powerful implications. Many extensions of the Abadi and Rogaway result simply rely on soundness as a ‘black-box’ assumption, and are not themselves hindered by key-cycles. By removing the key-cycle restriction from the Abadi-Rogaway result, it is removed from these extensions as well.

Consider, for example, the non-malleability results of Herzog [11]. In this setting, the adversary does not wish to distinguish two expressions but to transform one expression M into another expression M' . The formal adversary has only a limited power to do this, and can only produce formal messages in a set called the *closure* of M (denoted $C[M]$). Soundness for this non-malleability property is that no computational adversary, given the interpretation of M , can produce the interpretation of an expression outside $C[M]$. As Herzog shows, this soundness for this non-malleability property is directly implied by soundness for indistinguishability of messages (Definition 10). Because we show the KDM security soundness for message indistinguishability, this result of Herzog shows that it also provides soundness for non-malleability properties as well.

5.3 A Strictly New Notion

We now briefly prove 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 chosen-ciphertext security (CCA-2). The former is proved directly, Theorem 3, while the latter is a corollary to previous theorems:

Corollary 2. *CCA-2 security does not imply KDM-security. If there exists an encryption scheme secure against the chosen-ciphertext attack, there exists an encryption scheme which is secure against the chosen-ciphertext attack but not KDM-secure.*

Proof. Suppose that there exists a CCA-2 secure encryptions scheme. By Theorem 1 there is a CCA-2 secure scheme Π such that Π does not satisfy soundness. If all CCA-2 encryptions schemes are KDM-secure, then Π is as well. By Theorem 2, this means that Π satisfies soundness—a contradiction. \square

Theorem 3. *KDM security $\not\Rightarrow$ NM-CPA security. That is, there is an encryption scheme that is KDM-secure, but not NM-CPA secure.*

Proof. This is easily seen by inspecting the KDM-secure encryption scheme given by Black *et al.* in the random oracle model [7]. Let \mathcal{F} be a trapdoor permutation generator. Then:

- $\mathcal{K} = \mathcal{F}$ produces pairs (f, f^{-1}) where f encodes a trapdoor permutation and f^{-1} encodes its inverse,
- \mathcal{E} , on input (f, M) , selects a random bit-string r and returns the pair $(f(r), RO(r) \oplus M)$ (where RO is the random oracle),
- \mathcal{D} , on input $(f^{-1}, C = (c_1, c_2))$, returns $RO(f^{-1}(c_1)) \oplus c_2$.

This scheme is not NM-CPA secure: it is simple to change the ciphertext associated with a message M into the ciphertext of a related message. Note that an encryption of M provides confidentiality by essentially applying a random r as a one-time pad. Thus, changing a single bit of the (second component of a) ciphertext changes the same bit of the plaintext. That is, if $C = (f(r), RO(r) \oplus M)$ is an encryption of M , one can easily create $C' = (f(r), RO(r) \oplus \overline{M})$ (where \overline{M} is the bit-wise complement of M). C' decrypts to \overline{M} . Thus, this KDM-secure encryption scheme does not provide non-malleability of ciphertexts. \square

Due to the various relations among the security notions (see Appendix A) we have the following corollary:

Corollary 3. *KDM security implies neither NM-CCA1 security nor CCA2 security.*

Proof. Suppose that KDM implies NM-CCA1. Since NM-CCA1 implies NM-CPA we have that KDM implies NM-CPA—contradicting Theorem 3. Thus, KDM cannot imply NM-CCA1. Similarly for CCA2. \square

We conclude our discussion on the relationships between different notions of security by showing that soundness does not imply IND-CPA:

Theorem 4. *Soundness does not imply IND-CPA. That is, if there exists an encryption scheme with provides soundness, there exists a scheme which provides soundness but is not IND-CPA.*

Proof. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a sound encryption scheme. Let $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ be the following. Let $\mathcal{K}' = \mathcal{K}$. Let \mathcal{E}' do the same on an input of a pair of a public key and a plaintext (k, x) as \mathcal{E} for all plaintext, except when x is the security parameter given by k , in which case \mathcal{E}' outputs a fixed bit-string σ of the same length as $\mathcal{E}(k, x)$. \mathcal{D}' is the corresponding modified decryption algorithm.

This encryption scheme is still sound, because the interpretation of any expression with respect to \mathcal{E} is indistinguishable from the interpretation of this same expression with respect to \mathcal{E}' . The reason for this is the following: For each security parameter, there is only one string that is encrypted differently by \mathcal{E} and \mathcal{E}' . Let Φ and Φ' denote the respective interpretations. For any K public or private key, $\llbracket K \rrbracket_{\Phi} = \llbracket K \rrbracket_{\Phi'}$ trivially, and also $\llbracket B \rrbracket_{\Phi} = \llbracket B \rrbracket_{\Phi'}$ for any block B . Moreover these interpretations hit the security parameter with negligible probability. Now, for any expression M , if $\llbracket M \rrbracket_{\Phi} \approx \llbracket M \rrbracket_{\Phi'}$ and $\llbracket M \rrbracket_{\Phi'}$ hits the security parameter with negligible probability, then $\llbracket \{M\}_K \rrbracket_{\Phi} \approx \llbracket \{M\}_K \rrbracket_{\Phi'}$, and $\llbracket \{M\}_K \rrbracket_{\Phi'}$ hits the security parameter with

negligible probability. Similarly for pairing. Therefore, by induction, the two interpretations of a given expression are indistinguishable.

On the other hand, it is easy to see, that Π' is not IND-CPA secure, because the adversary in Definition 13 (in the Appendix) that submits the security parameter and 0^n to the oracles (that is, $m_0 = 0^n$, $m_1 = 1^n$) can certainly distinguish the two encryptions.

6 Conclusions

We have considered computational soundness of formal encryption. This property states that certain formal equivalence of symbolic expressions implies computational indistinguishability when the symbolic expressions are interpreted using the key generation and probabilistic encryption in a given computational encryption scheme. Computational soundness was proved in Abadi and Rogaway [1] under the assumption that there are no key cycles and that a computational encryption scheme satisfies a strong version of semantic security (so-called type-0 in the sense of Abadi and Rogaway [1]). We have considered a modification of the logic of formal encryption in the case of which-key revealing and message-length revealing, asymmetric encryption schemes (which corresponds to so-called type-3 in the sense of Abadi and Rogaway [1]). In the presence of key cycles, we have proved that the computational soundness property follows from the Key-Dependent Messages (KDM) security proposed by Black *et al.* [7]. As far as we know, this is the first time that in order to receive soundness, the computational model is strengthened and not the formal model weakened. We have also shown that the computational soundness property neither implies nor is it implied by security against chosen ciphertext attack, CCA-2. This is in contrast many previous results where CCA-2 is assumed to obtain computational soundness of symbolic analysis of authentication protocols.

Our work presents several directions for future research. First, they provide only more motivation to find an implementation of KDM security in the standard model (such as possibly Cramer-Shoup [?]). Secondly, it seems desirable to extend our results from the passive-adversary setting to that of the active adversary. Lastly, we show that the relationship between the formal and computational models requires more than chosen-ciphertext security. This investigation is not complete, and it is more than likely additional properties will be revealed as necessary as soundness is more fully explored.

References

1. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.
2. Pedro Adão, Gegei Bana, and Andre Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *18th IEEE Computer Security Foundations Workshop (CSFW 18)*, Aix-en-Provence, France, June 20 - 22 2005. IEEE Computer Society. To Appear.
3. Michael Backes and Birgit Pfizmann. Relating symbolic and cryptographic secrecy. *Cryptology ePrint Archive*, Report 2004/300, November 2004. <http://eprint.iacr.org/>.
4. Michael Backes, Birgit Pfizmann, and Michael Waidner. A composable cryptographic library with nested operations (extended abstract). In *Proceedings, 10th ACM conference on computer and communications security (CCS)*, October 2003. Full version available at <http://eprint.iacr.org/2003/015/>.
5. Gegei Bana. *Soundness and Completeness of Formal Logics of Symmetric Encryption*. PhD thesis, University of Pennsylvania, July 2004. Available at <http://www.math.upenn.edu/bana/banaphdthesis.pdf>.
6. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, August 1998. Full version found at <http://www.cs.ucsd.edu/users/mihir/papers/relations.html>.
7. John Black, Philip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75, St. John's, Newfoundland, Canada, August, 15-16 2002. Springer-Verlag.

8. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 98–118, Aarhus, Denmark, May 21–23 2001. Springer-Verlag.
9. Ran Canetti and Jonathan Herzog. Universally composable symbolic analysis of cryptographic protocols (the case of encryption-based mutual authentication and key exchange). Available at <http://eprint.iacr.org/2004/334>.
10. Vronique Cortier and Bogdan Warinschi. Computationally sound, automated proofs for security protocols. In Sagiv [22], pages 157–171.
11. Jonathan Herzog. *Computational Soundness for Standard Assumptions of Formal Cryptography*. PhD thesis, Massachusetts Institute of Technology, May 2004.
12. Jonathan Herzog, Moses Liskov, and Silvio Micali. Plaintext awareness via key registration. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 548–564. Springer-Verlag, August 2003.
13. Omer Horvitz and Virgil Gligor. Weak key authenticity and the computational completeness of formal encryption. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference*, volume 2729 of *Lecture Notes in Computer Science*, pages 530–547, Santa Barbara, California, USA, August, 17–21 2003. Springer-Verlag.
14. Romain Janvier, Yassine Lakhnech, and Laurent Mazar. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In Sagiv [22], pages 172–185.
15. Peeter Laud. Encryption cycles and two views of cryptography. In *NORDSEC 2002 – Proceedings of the 7th Nordic Workshop on Secure IT Systems*, number 31 in Karlstad University Studies, pages 85–100, Karlstad, Sweden, November, 7–8 2002.
16. Peeter Laud. Symmetric encryption in automatic analyses for confidentiality against active adversaries. In *IEEE Symposium on Security and Privacy*, pages 71–85, Oakland, CA, USA, May, 9–12 2004. IEEE Computer Society.
17. Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In Jong In Lim and Dong Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003: 6th International Conference*, volume 2971 of *Lecture Notes in Computer Science*, pages 55–66, Seoul, Korea, November, 27–28 2003. Springer-Verlag.
18. Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, April 1998.
19. Daniele Micciancio and Saurabh Panjwani. Adaptive security of symbolic encryption. In Joe Kilian, editor, *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 169–187, Cambridge, MA, USA, February 10–12 2005. Springer-Verlag.
20. Daniele Micciancio and Bogdan 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.
21. Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In Moni Naor, editor, *Proceedings, Theory of Cryptography Conference*, number 2951 in *Lecture Notes in Computer Science*, pages 133–151. Springer, February 2004.
22. Mooly Sagiv, editor. *Programming Languages and Systems: 14th European Symposium on Programming, ESOP 2005*, volume 3444 of *Lecture Notes in Computer Science*, Edinburgh, UK, April 4–8 2005. Springer-Verlag.
23. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science (FOCS 1999)*, pages 543–553. IEEE Computer Society, October 1999.
24. Bogdan Warinschi. A computational analysis of the Needham-Schroeder protocol. In *16th IEEE Computer Security Foundations Workshop (CSFW-16 2003)*, pages 248–262, Pacific Grove, CA, USA, June 30 - July 2 2003. IEEE Computer Society.

A Computational Definitions of Security For Asymmetric Encryption Schemes

We present the standard computational notions of security for asymmetric encryption schemes. See Figure 1 for their relationships.

A.1 Indistinguishability Notions

Definition 13 (IND-CPA — Chosen-Plaintext Security (Semantic Security)). *A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides indistinguishability under the chosen-plaintext attack, if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\Pr \left[\begin{array}{l} (e, d) \leftarrow \mathcal{K}(1^\eta); \\ m_0, m_1 \leftarrow A(1^\eta, e); i \leftarrow \{0, 1\}; \\ c \leftarrow \mathcal{E}(e, m_i); g \leftarrow A(1^\eta, e, c) : \\ b = g \end{array} \right] \leq \frac{1}{2} + \text{neg}(\eta)$$

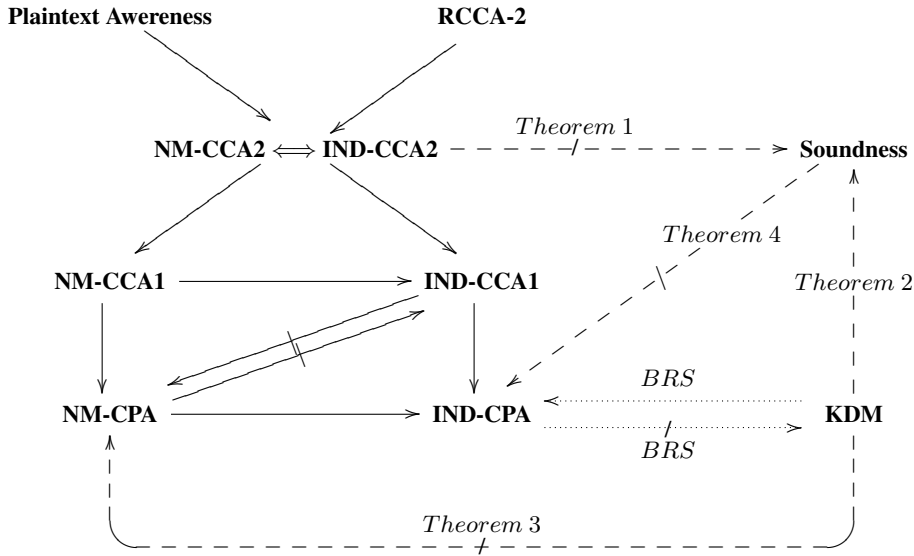


Fig. 1. Relation Among Different Security Notions

The adversary is assumed to keep state between the two invocations. It is required that m_0 and m_1 be of the same length.

Definition 14 (IND-CCA1 — Non-Adaptive Chosen-Ciphertext Security (Lunch-Time Security)). A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides indistinguishability under non-adaptive chosen-ciphertext attack if for all PPT adversaries A and for all sufficiently large security parameter η :

$$\Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); \\ m_0, m_1 \leftarrow \mathcal{A}^{\mathcal{D}_1(\cdot)}(1^\eta, e); i \leftarrow \{0, 1\}; \\ c \leftarrow \mathcal{E}(e, m_i); g \leftarrow A(1^\eta, e, c) : \\ b = g \quad \quad \quad] \leq \frac{1}{2} + \text{neg}(\eta)$$

The oracle $\mathcal{D}_1(x)$ returns $\mathcal{D}(d, x)$. The adversary is assumed to keep state between the two invocations. It is required that m_0 and m_1 be of the same length.

Definition 15 (IND-CCA2 — Adaptive Chosen-Ciphertext Security). A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides indistinguishability under the chosen-ciphertext attack if for all PPT adversaries A and for all sufficiently large security parameter η :

$$\Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); \\ m_0, m_1 \leftarrow \mathcal{A}^{\mathcal{D}_1(\cdot)}(1^\eta, e); i \leftarrow \{0, 1\}; \\ c \leftarrow \mathcal{E}(e, m_i); g \leftarrow \mathcal{A}^{\mathcal{D}_2(\cdot)}(1^\eta, e, c) : \\ b = g \quad \quad \quad] \leq \frac{1}{2} + \text{neg}(\eta)$$

The oracle $\mathcal{D}_1(x)$ returns $\mathcal{D}(d, x)$, and $\mathcal{D}_2(x)$ returns $\mathcal{D}(d, x)$ if $x \neq c$ and returns \perp otherwise. The adversary is assumed to keep state between the two invocations. It is required that m_0 and m_1 be of the same length.

A.2 Non-Malleability Notions

In a Non-Malleability game an adversary does not need to distinguish between two ciphertexts, in contrast with Indistinguishability notions. In the NM case, when an adversary has access to a ciphertext c , that is an encryption of a certain element m from a pre-defined message space M just composed by messages with the same length, he needs to output a relation R and a vector of ciphertexts \mathbf{c} such that the vector of decryptions \mathbf{m} has some relation with m , i.e., $R(m, m_i)$ for all i . The adversary wins the game if he can do this with a probability significantly better than that obtained from $R(\tilde{m}, m_i)$ for some random \tilde{m} also drawn from M . We will use $R(m, \mathbf{m})$ as a shorthand for $R(m, m_i)$ for all i .

Definition 16 (NM-CPA — Non-Malleability under Chosen-Plaintext Attacks). *A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides non-malleability under the chosen-plaintext attack if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\begin{aligned} & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(m, \mathbf{m})] - \\ & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(\tilde{m}, \mathbf{m})] \leq \text{neg}(\eta) \end{aligned}$$

We have that M is a valid message space, that is, $|m| = |m'|$ for any message m, m' that are given non-zero probability in the message space M . We also require that c is not in the vector \mathbf{c} and all the elements in \mathbf{c} are valid encryptions, that is $\mathcal{D}(d, c_i) \neq \perp$ for all i . The adversary is assumed to keep state between the two invocations.

Definition 17 (NM-CCA1 — Non-Malleability under Non-Adaptive Chosen-Ciphertext Attacks). *A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides non-malleability under non-adaptive chosen-ciphertext attack if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\begin{aligned} & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A^{\mathcal{D}_1(\cdot)}(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(m, \mathbf{m})] - \\ & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A^{\mathcal{D}_1(\cdot)}(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(\tilde{m}, \mathbf{m})] \leq \text{neg}(\eta) \end{aligned}$$

We have that M is a valid message space, that is, $|m| = |m'|$ for any message m, m' that are given non-zero probability in the message space M . We also require that c is not in the vector \mathbf{c} and all the elements in \mathbf{c} are valid encryptions, that is $\mathcal{D}(d, c_i) \neq \perp$ for all i . The oracle $\mathcal{D}_1(x)$ returns $\mathcal{D}(d, x)$. The adversary is assumed to keep state between the two invocations.

Definition 18 (NM-CCA2 — Non-Malleability under Adaptive Chosen Ciphertext Attacks). *A computational public-key encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides non-malleability under non-adaptive chosen-ciphertext attack if for all PPT adversaries A and for all sufficiently large security parameter η :*

$$\begin{aligned} & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A^{\mathcal{D}_1(\cdot)}(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A^{\mathcal{D}_2(\cdot)}(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(m, \mathbf{m})] - \\ & \Pr[(e, d) \leftarrow \mathcal{K}(1^\eta); M \leftarrow A^{\mathcal{D}_1(\cdot)}(1^\eta, e); m \leftarrow M; \\ & \quad c \leftarrow \mathcal{E}(e, m); (R, \mathbf{c}) \leftarrow A^{\mathcal{D}_2(\cdot)}(1^\eta, e, M, c); \mathbf{m} \leftarrow \mathcal{D}(d, \mathbf{c}) : R(\tilde{m}, \mathbf{m})] \leq \text{neg}(\eta) \end{aligned}$$

We have that M is a valid message space, that is, $|m| = |m'|$ for any message m, m' that are given non-zero probability in the message space M . We also require that c is not in the vector \mathbf{c} and all the elements in \mathbf{c} are valid encryptions, that is $\mathcal{D}(d, c_i) \neq \perp$ for all i . The oracle $\mathcal{D}_1(x)$ returns $\mathcal{D}(d, x)$, and $\mathcal{D}_2(x)$ returns $\mathcal{D}(d, x)$ if $x \neq c$ and returns \perp otherwise. The adversary is assumed to keep state between the two invocations.

B Interpretation Algorithm

Let M be an expression.

algorithm $INITIALIZE_\eta(M)$
for $K \in Keys(M)$ **do** $(\tau(K), \tau(K^{-1})) \leftarrow \mathcal{K}(1^\eta)$

algorithm $CONVERT_\eta(M)$
if $M = K$ where $K \in \mathbf{Keys}$ **then**
 return $\tau(K)$
if $M = K^{-1}$ where $K \in \mathbf{Keys}^{-1}$ **then**
 return $\tau(K^{-1})$
if $M = B$ where $B \in \mathbf{Blocks}$ **then**
 return B
if $M = (M_1, M_2)$ **then**
 $x \leftarrow CONVERT_\eta(M_1)$
 $y \leftarrow CONVERT_\eta(M_2)$
 return $[x, y]$
if $M = \{M_1\}_K$ **then**
 $x \leftarrow CONVERT_\eta(M_1)$
 $y \leftarrow \mathcal{E}(\tau(K), x)$
 return y

For a pattern M we define the interpretation as

algorithm $INITIALIZE_\eta(M)$
for $K \in Keys(M)$ **do** $(\tau(K), \tau(K^{-1})) \leftarrow \mathcal{K}(1^\eta)$

algorithm $CONVERT_\eta(M)$
if $M = K$ where $K \in \mathbf{Keys}$ **then**
 return $\tau(K)$
if $M = K^{-1}$ where $K \in \mathbf{Keys}^{-1}$ **then**
 return $\tau(K^{-1})$
if $M = B$ where $B \in \mathbf{Blocks}$ **then**
 return B
if $M = (M_1, M_2)$ **then**
 $x \leftarrow CONVERT_\eta(M_1)$
 $y \leftarrow CONVERT_\eta(M_2)$
 return $[x, y]$
if $M = \{M_1\}_K$ **then**
 $x \leftarrow CONVERT_\eta(M_1)$
 $y \leftarrow \mathcal{E}(\tau(K), x)$
 return y

iiiiiii circuit **if** $M = \square_{K, \ell(M')}$, **then**
 $y \leftarrow \mathcal{E}(\tau(K), 0^{|\Phi_\eta(M')|})$

===== **if** $M = \square_{K, \ell(M')}$, **then**
 $y \leftarrow \mathcal{E}(\tau(K), 0^{|\Phi_\eta(M')|})$

iiiiiii 1.119 **return** y