Soundness of Formal Encryption in the Presence of Key-Cycles

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

[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 the computational models of cryptography contain the notion of message *equivalence* or *indistinguishability*. An encryption scheme provides *soundness* for indistinguishability if, when mapping formal messages into the computational model, equivalent formal messages are mapped to 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 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 a new notion of computational security, not possible using previous such notions, and the relationship between the formal and computational models extends beyond 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

⋆ Partially supported by FCT grant SFRH/BD/8148/2002. Additional support from FEDER/FCT project Fiblog POCTI/2001/MAT/37239 and FEDER/FCT project QuantLog POCI/MAT/55796/2004.

⋆⋆ 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.

⋆⋆⋆ 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.

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, we consider the relationship between these two models; more precisely, the relationship between a simplified formal model following the technique of Abadi and Rogaway, and the computational implementation of this model. 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 probability distributions over bit-strings (indexed by the security parameter). The adversary is modeled as an 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 of Abadi and Rogaway, 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 establishes the relationship 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, we say that *soundness* holds[4] and it implies that the formal model is a faithful abstraction of the computational model in the sense that security of the formal model implies security in the computational.

The first soundness result of this type is due to Abadi and Rogaway in the symmetric-key encryption setting [2]. They demonstrated that soundness holds when the security level of the computational encryption algorithm is 'type-0,' a property of their own

[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.

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 [41]. 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 [13]). This power of chosen-ciphertext security has been confirmed by subsequent extensions [29, 19]. 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 $\{M_1\}_{K_1}, \{M_2\}_{K_2}, \ldots, \{M_n\}_{K_n}$ (where $\{M_i\}_{K_i}$ denotes the encryption of the message M_i with the public key K_i) such that 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 those messages that posses key-cycles and those that do not. Further, 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 [10, 19] 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 [38]. The formal model, on the other hand, assumes that key-cycles do not weaken encryption in any way. Therefore, the issue of key-cycles may represent a ‘gap’ between the formal and computational models, and thus might shed light on their general relationship.

Gaps between the two models. The majority of the results relating the two models 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. For example, the original soundness results of Abadi and Rogaway assumed that formal encryption concealed all aspects of the plaintext. In particular, their result requires that symmetric encryption hides (among other things)the length of the plaintext. Unfortunately, this cannot be achieved for many contexts. Soundness in these other contexts is considered by Micciancio and Warinschi [41], Laud [34], Bana [11], Micciancio and Panjwani [39] and Adão, Bana and Scedrov [3], who require a weaker notion of formal equivalence. (In keeping with this, we will use the 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 one. Canetti and Herzog [19] and Backes and Pfitzmann [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 abstracts 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 onto the formal model. Should the resolution of the problem of key-cicles again cause changes to the formal model, or could it this time be more naturally resolved through modifications to the computational model?

An alternate approach. Laud [33] has proposed a solution to the problem of key-cycles which takes the first approach. That is, 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 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 simultaneously both by Black, Rogaway and Shrimpton [14], who consider it in their own right, and by Camenisch and Lysyanskaya [16], 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. Also, 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 with recent advances in the computational model it can be repaired. Also, soundness in the presence of

[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 [30] for fuller discussion and an alternate definition.

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.

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 in the language of Abadi and Rogaway [2], 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 does 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 because 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 also show that soundness does not imply semantic security (IND-CPA security, in the notation of [13]).

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

Acknowledgements. This work was done while the first author was a visiting student at the University of Pennsylvania. We want to thank J. Black, A. Gordon, S. Hohenberger, A. Lysyanskaya, T. Shrimpton, and D. Unruh 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 also thank the anonymous referees for their extensive comments and suggestions.

2 Previous Work

Work intended to bridge the gap between the cryptographic and the formal models started with several independent approaches, including Lincoln, Mitchell, Mitchell, and Scedrov [36], Canetti [18], Pfitzmann, Schunter and Waidner [43, 44], and Abadi and Rogaway [2]. In [2], 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 [2] further below. Several papers consider specific models or specific properties, *e.g.,* Guttman, Thayer, and Zuck [26] specifically 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 [36]. In this framework, which provides a formal treatment of the computational model, security properties are formulated as observational equivalences. Mitchell, Ramanathan, Scedrov, and Teague [42] 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 [43, 44] 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 [27, 37, 15, 17]. The approach was extended from synchronous to asynchronous systems in [45, 18], which are now known as the *reactive simulatability framework* [45, 8] and the *universal composability framework* [18]. A detailed comparison of the two approaches may be found in [23].

The first soundness result of a formal model under active attacks has been achieved by Backes, Pfitzmann and Waidner [10] 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 [4, 9]. While the original result in [10] considered public-key encryption and digital signatures, the soundness result was extended to symmetric authentication and to symmetric encryption in [7] and [6], respectively.

Concurrently with [10], an extension to asymmetric encryption, but still under passive attacks, is in [30]. Asymmetric encryption under active attacks is considered in [28] in the random oracle model. Laud [34] 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.* [30] and Micciancio and Warinschi [41] also give a cryptographic underpinning under active attacks. Their results are narrower than that in [10] since they are specific for public-key encryption, but consider simpler real implementations. Moreover, [30] relies on a stronger assumption, which was subsequently weakened by Herzog [29]. The approach in [41] restricts the classes of protocols and protocol properties that can be analyzed. The work of [41] was subsequently extended by Micciancio and Panjwani [39] to prove soundness of a group-key distribution protocol in the presence of a CPA-secure scheme. Cortier and Warinschi [21] 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 [11] and Adão, Bana, and Scedrov [3] extend the original Abadi-Rogaway result to weaker encryption schemes. Laud and Corin [35] consider extensions to composite keys, while Baudet, Cortier, and Kremer [12] consider extensions to equational theories and to static equivalence.

Impagliazzo and Kapron [32] suggest a formal logic for reasoning about probabilistic polynomial-time indistinguishability. Datta, Derek, Mitchell, Shmatikov, and Turuani [24] 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 [21] 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 [19] 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 pointed out a sufficient symbolic criteria to derive if a payload is cryptographically secret.

The first cryptographically sound security proofs of the Needham-Schroeder-Lowe protocol have been presented concurrently and independently in [5] and [47]. 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.

The relation 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 [40] show that a sufficiently strong encryption scheme enforces completeness for indistinguishability properties, and later Horvitz and Gligor [31] 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 [11] and Adão, Bana, and Scedrov [3] that completeness also holds for a more general class of (weaker) encryption systems. We only briefly mention that the simulatability-based results of [10, 7, 6] have shown completeness implicitly to establish the notion of simulatability. We do not discuss completeness any further in this work.

Finally, we stress that none of the aforementioned results hold in the presence of key-cycles. As we mentioned in the introduction, this problem was addressed by Laud [33] in a different way from the one that we will address in this paper.

3 Computational Soundness for Indistinguishability

We start presenting the formal model, and then describe the computational model in a fairly standard way. Then, 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 identical to the treatment of Abadi and Rogaway [2], with three exceptions: 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 **Keys** $= \{K_1, K_2, K_3, ...\}$ be an infinite discrete set of symbols, called the set of encryption keys, and **Keys**$^{-1} = \{K_1^{-1}, K_2^{-1}, K_3^{-1}, ...\}$ the corresponding set of decryption keys. Let **Blocks** be a finite subset of $\{0,1\}^*$. We define the* set of expressions*, **Exp**, by the grammar:*

$$\textbf{\textit{Exp}} ::= \textbf{\textit{Keys}} \quad | \quad \textbf{\textit{Keys}}^{-1} \quad | \quad \textbf{\textit{Blocks}} \quad | \quad (\textbf{\textit{Exp}}, \textbf{\textit{Exp}}) \quad | \quad \{\textbf{\textit{Exp}}\}_{\textbf{\textit{Keys}}}$$

We will denote by Keys(M) the set of all encryption keys occurring in M and by 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. 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 can then 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 (Visible Subterms, Recoverable Decryption Keys). *Let $vis\,(M) \subseteq$ **Exp**, the* visible subterms of M*, be the smallest set of expressions containing M such that:*

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

Let R-Keys(M), the set of recoverable decryption keys *in M, be $vis\,(M) \cap$ **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 [40, 29, 11]:

Definition 3 (Formal Length). *We introduce a function symbol with fresh letter ℓ with the following identities:*

- *For all blocks B_1 and B_2, $\ell(B_1) = \ell(B_2)$ iff $|B_1| = |B_2|$,*
- $\forall i, j \in \mathbb{N}$, $\ell(K_i) = \ell(K_j)$ *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 express that the encryption operation may leak information about the length. We note that when **Blocks** is just $\{0, 1\}$, then equality of $\ell(M)$ and $\ell(N)$ implies that M and N have identical type trees.

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 formal equivalence of messages. This requires us to define what is 'observable' for an adversary in an expression. In order to express that, we define the so-called *pattern* of an expression, and two expressions will be considered equivalent when their patterns are (roughly speaking) identical:

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

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

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 $K^{-1} \notin$ R-Keys(M)) by $\square_{K,\ell(M')}$.

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

- *If $P \in$ **Blocks** $\cup$ **Keys** $\cup$ **Keys**$^{-1}$, then $P = Q$ iff P and Q are identical.*
- *If P is of the form $\square_{K,\ell(M')}$, then $P = Q$ iff Q is of the form $\square_{K,\ell(N')}$, and $\ell(M') = \ell(N')$ in the sense of Definition 3.*
- *If P is of the form (P_1, P_2), then $P = Q$ iff Q is of the form (Q_1, Q_2) where $P_1 = P_2$ and $Q_1 = Q_2$.*
- *If P is of the form $\{P'\}_K$, then $P = Q$ iff Q is of the form $\{Q'\}_K$ where $P' = Q'$.*

The symbol $\square_{K,\ell(M')}$ in a pattern reveals that some expression 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. Consider two formal expressions that differ only in the names of the keys in them, but such that if there are identical keys in one of them, there are corresponding identical keys in the other in the same place. On the other hand, two keys, say, K_1 and K_2, have the same meaning: a randomly drawn key, using the same key-generation algorithm. It does not matter if we replace one of them with the other. The appearance of a new key in an expression just means a freshly generated key, it does not matter what name we give it. What matters is only where the identical keys are in an expression, and where are the differing ones. 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 5 (Key-Renaming Function). *A bijection σ : **Keys** $\to$ **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 pattern$(M) =$ pattern$(N\sigma)$.*

Our main focus in this paper is on key-cycles:

Definition 7 (Key-Cycles). *A formal message M contains a* key-cycle *if it contains encryption terms $\{M_1\}_{K_1}$, $\{M_2\}_{K_2}$, ..., $\{M_n\}_{K_n}$ (where $\{M_i\}_{K_i}$ denotes the encryption of the message M_i with the public key K_i) such that M_i contains the key necessary to decrypt $\{M_{i+1}\}_{K_{i+1}}$ and M_n contains the key necessary to decrypt $\{M_1\}_{K_1}$. In this case we say that we have a key-cycle of length n.*

3.2 The Computational Model

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

Definition 8 (Negligible Function). *A function $f : \mathbb{N} \to \mathbb{R}$ is said to be* negligible*, written $f(n) \leq \text{neg}(n)$, 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*,*

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

In this model, pairing is an injective *pairing function* $[\cdot,\cdot]$: $\mathsf{strings} \times \mathsf{strings} \to \mathsf{strings}$ 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 $\mathsf{plaintexts}$, $\mathsf{ciphertexts}$, $\mathsf{publickey}$ and $\mathsf{secretkey}$ be nonempty subsets of $\mathsf{strings}$. The set coins is some probability field that stands for coin-tossing, *i.e.*, randomness.

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

- $\mathcal{K}$: $\mathsf{parameters} \times \mathsf{coins} \to \mathsf{publickey} \times \mathsf{secretkey}$ *is a key-generation algorithm with security parameter η,*
- $\mathcal{E}$: $\mathsf{publickey} \times \mathsf{plaintexts} \times \mathsf{coins} \to \mathsf{ciphertexts}$ *is an encryption function, and*
- $\mathcal{D}$: $\mathsf{secretkey} \times \mathsf{strings} \to \mathsf{plaintexts}$ *is such that for all $(e, d) \in \mathsf{publickey} \times \mathsf{secretkey}$ and $\omega \in \mathsf{coins}$*

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

All these algorithms must be computable in polynomial time in the size of the input not counting the coins*. (For this reason, the set* $\mathsf{parameters}$ *is usually represented as 1^*.) We insist that $|\mathcal{E}(e, m, w)| = |\mathcal{E}(e, m, w')|$ for all $e \in \mathsf{publickey}, m \in \mathsf{plaintexts}$ and $w, w' \in \mathsf{coins}$, where $|x|$ stands for the binary length of x. We also insist that $0^* \subseteq \mathsf{plaintexts}$. We lastly insist that for all e and 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 $(e, d) \longleftarrow \mathcal{K}(1^\eta)$).*

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 $\mathsf{strings}$. As in Abadi and Rogaway [2], this interpretation is defined in an algorithmic way. The full formalism is given in Appendix B, but we present an informal overview here. For expressions:

- Blocks are interpreted as $\mathsf{strings}$,
- 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 $[\![M]\!]_{\Phi_\eta}$ the distribution of $\Phi_\eta(M)$ and by $[\![M]\!]_\Phi$ the ensemble of $\{[\![M]\!]_{\Phi_\eta}\}_{\eta\in\mathbb{N}}$. For the interpretation of patterns, everything is the same as for the interpretation of expressions, but we also have:

– The interpretation of a pattern $\Box_{K,\ell(M)}$ for a given security parameter η is given by $\Phi_\eta(\{0^{|\Phi_\eta(M)|}\}_K)$ where $|\Phi_\eta(M)|$ is the binary length of $\Phi_\eta(M)$, which must be the same for all samples (due to our assumptions about 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, $[\![M]\!]_{\Phi_\eta}$ the distribution of $\Phi_\eta(M)$ and $[\![M]\!]_\Phi$ the ensemble of distributions $\{[\![M]\!]_{\Phi_\eta}\}_{\eta\in\mathbb{N}}$.

We can now define the notion of soundness.

Definition 11 (Soundness). *We say that an interpretation is* sound*, or that an encryption scheme* provides soundness*, if the interpretation Φ (resulting from the encryption scheme) is such that*

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

for any expressions M and N.

The primary result of Abadi and Rogaway given in [2] 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 [41] 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 [29, 19] confirms that chosen-ciphertext security suffices for several extensions, so long as key-cycles are prohibited. In the next section, we show that this prohibition was necessary: in the presence of key-cycles, chosen-ciphertext does not necessarily guarantee soundness.

4 Chosen-Ciphertext Security is Not Enough

In this section we show that these notions of security, which were standard when the results of Abadi and Rogaway were published, are 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 (in particular, CCA-2 in the notation of [13]) but fail to provide soundness in the presence of key-cycles.

Definition 12 (IND-CCA2—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 parameters η:*

$$\begin{aligned}\Pr[\ &(e,d) \longleftarrow \mathcal{K}(1^\eta);\\ &m_0, m_1 \longleftarrow \mathsf{A}^{\mathcal{D}_1(\cdot)}(1^\eta, e);\\ &i \longleftarrow \{0,1\}\,;\\ &c \longleftarrow \mathcal{E}(e, m_i);\\ &g \longleftarrow \mathsf{A}^{\mathcal{D}_2(\cdot)}(1^\eta, e, c) :\\ &b = g \qquad\qquad\qquad\qquad] \leq \tfrac{1}{2} + \operatorname{neg}(\eta)\end{aligned}$$

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 [13]. It does not, however, guarantee soundness: A does not have (obviously) access to the private keys, and therefore the messages submitted to the oracles $\mathcal{D}_1$ and $\mathcal{D}_2$ cannot depend on those private keys. Therefore key-dependent messages are not considered and 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.*

We motivate the proof with a simple example: one-time pads. Although this is a form of symmetric encryption and the rest of this paper discusses asymmetric encryption, the main ideas translate:

Example 1 (One-Time Pad). Consider a key-cycle of length 1, such as the expression $M = \{K\}_K$. When interpreted using one-time pads, $[\![M]\!]_\Phi$ will become a sequence of elements from 0^*. However, we note that M is equivalent to the expression $N = \{K'\}_K$, yet the interpretation of N will be a family of uniformly random distributions. Thus, two equivalent expressions yield easily distinguished distribution families.

A similar argument, using CCA-2 encryption schemes instead of one-time pads, will suffice to prove Theorem 1. Given a CCA-2 secure encryption scheme, another CCA-2 encryption scheme is constructed which will provide distinguishable interpretations for expressions M and N above.

Proof. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a CCA-2 secure encryption scheme. We construct a second CCA-2 secure encryption scheme $\Pi' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ such that $\mathcal{K}' = \mathcal{K}$, $\mathcal{D}' = \mathcal{D}$, and $\mathcal{E}'$ is as follows:

- 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 $c \longleftarrow \mathcal{E}(e, r)$;
  - Let $p \longleftarrow \mathcal{D}(m, c)$;
  - Test whether $p = r$.
- If m is the decryption key associated with e, output m;
- Otherwise, compute $c' \longleftarrow \mathcal{E}(e, m)$ and output c'.

Since Π' acts exactly like Π when plaintexts and encryption keys are unrelated, Π' must be also CCA-2 secure. However, Π' cannot be KDM-secure. Let M be the formal expression $\{K^{-1}\}_K$, and let N be the expression $\{K'^{-1}\}_K$. These two expressions are equivalent, but $[\![M]\!]_\Phi$ can be easily distinguished from $[\![N]\!]_\Phi$: the first distribution family will output a decryption key while the second outputs a ciphertext. $\square$

Remark 2. We note that in both the example and the proof, the expression M contains a key-cycle of length 1. What if all key-cycles are of length 2 or more? The one-time pad still fails to provide soundness: the interpretation of $(\{K_1\}_{K_2}, \{K_2\}_{K_1})$ is a pair of completely correlated distributions, while the interpretation of $(\{K_1\}_{K_2}, \{K_3\}_{K_1})$ is a pair of independent distributions. The same question in the public-key setting, however, remains open. That is, there is no known CCA-2 secure encryption scheme which fails to provide soundness for key-cycles that are of length two or more.

Since CCA-2 security implies a number of other definitions [13] (see the figure in 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 standardcomputational 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, *key-dependent message* (KDM) security, which was introduced by Black *et al.* [14] (and in a weaker form by Camenisch and Lysyanskaya [16]), 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 strengthens IND-CPA (semantic) security, a weaker form of Definition 12 in which the adversary does not have access to the decryption oracles. However, semantic security still allows the adversary to submit two messages to be encrypted. KDM strengthens this by allowing more general submissions. 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 $\mathsf{Real}_{\mathbf{d}}$ and $\mathsf{Fake}_{\mathbf{d}}$, which work as follows:

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

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

Definition 13 (KDM Security). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme. Let the two oracles* $\mathsf{Real}_{\mathbf{d}}$ *and* $\mathsf{Fake}_{\mathbf{d}}$ *be as defined above. We say that the encryption scheme is* KDM-secure *if for all PPT adversaries* A *and for all sufficiently large security parameters η:*

$$\Big|\Pr\big[(\mathbf{e}, \mathbf{d}) \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathsf{Real}_{\mathbf{d}}}(1^\eta, \mathbf{e}) = 1\big] - \Pr\big[(\mathbf{e}, \mathbf{d}) \longleftarrow \mathcal{K}(1^\eta) : \mathsf{A}^{\mathsf{Fake}_{\mathbf{d}}}(1^\eta, \mathbf{e}) = 1\big]\Big| \leq \operatorname{neg}(\eta)$$

Remark 3. We note that although all known implementations of KDM-security are in the random-oracle model, this definition is well-founded even in the standard model. We also note that this definition is phrased in terms of indistinguishability. One could also imagine analogous definitions phrased in terms of non-malleability, but an exploration of those are beyond the scope of the paper.

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. If Π is KDM-secure, then Π provides 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 $[\![M]\!]_\Phi$ is indistinguishable from $[\![\mathit{pattern}(M)]\!]_\Phi$. Then, since that holds for N too, and since *pattern*(M) differs from *pattern*(N) only in the name of keys, $[\![\mathit{pattern}(M)]\!]_\Phi$ is indistinguishable from $[\![\mathit{pattern}(N)]\!]_\Phi$, therefore the result follows. KDM security is used when we show that $[\![M]\!]_\Phi$ and $[\![\mathit{pattern}(M)]\!]_\Phi$ 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 $\mathsf{publickey}$ and $\tau|_{\mathbf{Keys}^{-1}}$ takes values in $\mathsf{secretkey}$, we define a function $f_{M,S,\tau} : \mathsf{coins}^{e(M)} \times \mathsf{secretkey}^\infty \to \mathsf{strings}$ (where $e(M)$ is the number of encryptions in M) inductively the following way:

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

We first prove that $[\![M]\!]_\Phi \approx [\![\mathit{pattern}(M)]\!]_\Phi$. Suppose that $[\![M]\!]_\Phi \not\approx [\![\mathit{pattern}(M)]\!]_\Phi$, which means that there is an adversary A that distinguishes the two distributions, that is

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

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 $\mathsf{Real}_\mathbf{d}$ and $\mathsf{Fake}_\mathbf{d}$. Let $\mathcal{F}$ denote either of these oracles. Let $\mathbf{e} \in \mathsf{publickey}^\infty$ be the array of public keys that $\mathcal{F}$ outputs. From now on, let $S = \mathbf{Keys}^{-1} \setminus \mathit{R\text{-}Keys}(M)$, 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 $K^{-1} \in \mathit{R\text{-}Keys}(M)$, do $(\tau(K), \tau(K^{-1})) \longleftarrow \mathcal{K}(1^\eta)$
    $y \longleftarrow \mathrm{CONVERT2}_\mathbf{e}(M, M)$
    $b \longleftarrow \mathsf{A}(1^\eta, y)$
    return b

algorithm CONVERT2$_\mathbf{e}(M', M)$ with $M' \sqsubseteq M$
    if $M' = K$ where $K \in$ **Keys then**
        return $\tau(K)$
    if $M' = K^{-1}$ where $K^{-1} \in \mathit{R\text{-}Keys}(M)$ **then**
        return $\tau(K^{-1})$
    if $M = B$ where $B \in$ **Blocks then**

return B
if $M' = (M_1, M_2)$ **then**
$x \longleftarrow \text{CONVERT2}_{\mathbf{e}}(M_1, M)$
$y \longleftarrow \text{CONVERT2}_{\mathbf{e}}(M_2, M)$
return $[x, y]$
if $M' = \{M''\}_K$ with $K^{-1} \in \textit{R-Keys}(M)$ **then**
$x \longleftarrow \text{CONVERT2}_{\mathbf{e}}(M'', M)$
$y \longleftarrow \mathcal{E}(\tau(K), x)$
return y
if $M' = \{M''\}_K$ with $K^{-1} \notin \textit{R-Keys}(M)$ **then**
$\omega \longleftarrow \mathsf{coins}^{e(M'')}$
$y \longleftarrow \mathcal{F}(\iota(K), f_{M'',S,\tau}(\omega, .))$
return y

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

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

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 [29]. 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 11). 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 provide brief propositions about what Black *et al.* claimed informally: the notion of KDM security is 'orthogonal' to the previous definitions of security. In particular, we claim 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.*

Theorem 3. *KDM security does not imply 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 [14]. 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,
- The encryption algorithm $\mathcal{E}$, 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 $\left(f^{-1}, C = (c_1, c_2)\right)$, returns $RO\left(f^{-1}(c_1)\right) \oplus c_2$.

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 $C' = \left(f(r), RO(r) \oplus \overline{M}\right)$ (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. □

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.*

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, $[\![K]\!]_{\Phi} = [\![K]\!]_{\Phi'}$ trivially, and also $[\![B]\!]_{\Phi} = [\![B]\!]_{\Phi'}$ for any block B. Moreover these interpretations hit the security parameter with negligible probability. Now, for any expression M, if $[\![M]\!]_{\Phi} \approx [\![M]\!]_{\Phi'}$ and $[\![M]\!]_{\Phi'}$ hits the security parameter with negligible probability, then $[\![\{M\}_K]\!]_{\Phi} \approx [\![\{M\}_K]\!]_{\Phi'}$, and $[\![\{M\}_K]\!]_{\Phi'}$ 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 an adversary who submits as candidate messages the security parameter and 0^η (that is, outputs $m_0 = 0^\eta$, $m_1 = 1^\eta$) will certainly be able to determine which of the two messages was encrypted.

These statements are summarized in a figure in Appendix A.

6 Conclusions

We have considered computational soundness of formal encryption. This property states that formal equivalence of symbolic expressions implies computational indistinguishability when the symbolic expressions are interpreted using a given computational encryption scheme. Computational soundness was proved in Abadi and Rogaway [2] 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 [2]). We have considered a modification of their logic 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 [2]). 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.* [14]. As far as we know, this is the first time that in order to achieve 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 to many previous results where forms of soundness are implied by CCA-2 security.

Our work presents several directions for future research. Firstly, several questions about KDM security (independently of any soundness considerations) remain unanswered. An implementation of KDM security in the standard model remains to be found, although there are several natural candidates (for instance Cramer-Shoup [22]). Conversely, there remains to be found a natural (*i.e.,* non-constructed) example of an encryption scheme which is secure in the sense of CCA-2 but is not KDM-secure. Further, the constructed examples of such encryption schemes only fail to provide KDM security when presented with key-cycles of length 1. It may be possible that CCA-2 security implies KDM security when all key-cycles are of length 2 or more. Lastly, similar questions can also be posed in the setting of symmetric-key encryption—a course of investigation we are currently investigating.

With regard to soundness, on the other hand, it seems desirable to extend our results from the passive-adversary setting to that of the active adversary. Also, we show that the relationship between the formal and computational models requires more than chosen-ciphertext 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. That is, this investigation is not complete; it is more than likely additional properties will be revealed as necessary as soundness is more fully explored.

References

1. M. Abadi and J. Jürjens. Formal eavesdropping and its computational interpretation. In *Proc. 4-th International Symp. on Theor. Aspects of Comp. Software (TACS)*, Springer LNCS Vol. 2215, pp. 82–94, 2001.
2. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002. Prelim. version in IFIP TCS'00.
3. P. Adão, G. Bana, and A. Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *Proc. 18-th IEEE Computer Security Foundations Workshop (CSFW)*, pp. 170–184, IEEE Comp. Soc. Press, 2005.
4. M. Backes and C. Jacobi. Cryptographically sound and machine-assisted verification of security protocols. In *Proc. 20-th Annual Symp. on Theor. Aspects of Comp. Sci. (STACS)*, Springer LNCS Vol. 2607, pp. 675–686, 2003.
5. M. Backes and B. Pfitzmann. A cryptographically sound security proof of the Needham-Schröeder-Lowe public-key protocol. *IEEE J. Selected Areas in Communications*, 22(10):2075–2086, 2004. Prelim. version in FSTTCS'03.
6. M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable Dolev-Yao style cryptographic library. In *Proc. 17-th IEEE Computer Security Foundations Workshop (CSFW)*, 2004. Full version on ePrint 2004/059.
7. M. Backes, B. Pfitzmann, and M. Waidner. Symmetric authentication within a simulatable cryptographic library. In *Proc. 8-th European Symp. on Research in Comp. Security (ESORICS)*, Springer LNCS Vol. 2808, pp. 271–290, 2003. Extended version on ePrint 2003/145.
8. M. Backes, B. Pfitzmann, and M. Waidner. Secure asynchronous reactive systems. ePrint 2004/082.
9. M. Backes and B. Pfitzmann. Relating symbolic and cryptographic secrecy. *IEEE Trans. on Dependable and Secure Computing*, 2(2):109–123, 2005. Full version on ePrint 2004/300.
10. M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *Proc. 10-th ACM Conf. on Computer and Communications Security (CCS)*, pp. 220–230, ACM Press, 2003. Full version on ePrint 2003/015.
11. G. Bana. *Soundness and Completeness of Formal Logics of Symmetric Encryption*. PhD thesis, University of Pennsylvania, 2004. Available on ePrint 2005/101.
12. M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In: ICALP'05, Springer LNCS Vol. 3580, pp. 652–663, to appear.
13. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In CRYPTO '98, Springer LNCS Vol. 1462, pp. 26–45, 1998. Full version available at `http://www.cs.ucsd.edu/users/mihir/papers/relations.html`.
14. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Proc. 9-th Annual International Workshop on Selected Areas in Cryptography (SAC)*, Springer LNCS Vol. 2595, pp. 62–75, 2002.
15. D. Beaver. Secure multiparty protocols and zero knowledge proof systems tolerating a faulty minority. *J. Cryptology*, 4(2):75–122, 1991.

16. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, Springer LNCS Vol. 2045, pp. 98–118, 2001.
17. R. Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 3(1):143–202, 2000.
18. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42-nd IEEE Symp. on Foundations of Comp. Sci. (FOCS)*, pp. 136–145, IEEE Comp. Soc. Press, 2001. Full version on ePrint 2000/067.
19. R. Canetti and J. Herzog. Universally composable symbolic analysis of cryptographic protocols (the case of encryption-based mutual authentication and key exchange). ePrint 2004/334.
20. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proc. 30-th Annual ACM Symp. on Theory of Computing (STOC)*, pp. 209–218, ACM Press, 1998.
21. V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *Proc. 14-th European Symp. on Programming (ESOP)*, Springer LNCS Vol. 3444, pp. 157–171, 2005.
22. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO '98*, Springer LNCS Vol. 1462, pp. 13–25, 1998.
23. A. Datta, R. Küsters, J. C. Mitchell, and A. Ramanathan. On the relationships between notions of simulation-based security. In *2-nd Theory of Cryptography Conference, TCC 2005*, Springer LNCS Vol. 3378, pp. 476–494, 2005.
24. A. Datta, A. Derek, J. C. Mitchell, V. Shmatikov, and M. Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In ICALP'05, Springer LNCS Vol. 3580, to appear.
25. D. Dolev and A. C. Yao. On the security of public-key protocols. *IEEE Trans. on Information Theory*, 29(2):198–208, 1983. Prelim. version in FOCS'81.
26. J. D. Guttman, F. J. Thayer, and L. D. Zuck. The faithfulness of abstract protocol analysis: Message authentication. In *Proc. 8-th ACM Conf. on Computer and Communications Security (CCS)*, pp. 186–195, ACM Press, 2001.
27. S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In *CRYPTO '90*, Springer LNCS Vol. 537, pp. 77–93, 1990.
28. J. Herzog. *Computational Soundness of Formal Adversaries*. Master thesis, MIT, 2002.
29. J. Herzog. *Computational Soundness for Standard Assumptions of Formal Cryptography*. PhD thesis, MIT, 2004. Available at `http://theory.lcs.mit.edu/~jherzog/papers/herzog-phd.pdf`.
30. J. Herzog, M. Liskov, and S. Micali. Plaintext awareness via key registration. In *CRYPTO 2003*, Springer LNCS Vol. 2729, pp. 548–564, 2003.
31. O. Horvitz and V. Gligor. Weak key authenticity and the computational completeness of formal encryption. In *CRYPTO 2003*, Springer LNCS Vol. 2729, pp. 530–547, Sant 2003.
32. R. Impagliazzo and B. M. Kapron. Logics for reasoning about cryptographic constructions. In *Proc. 44-th IEEE Symp. on Foundations of Comp. Sci. (FOCS)*, pp. 372–381, IEEE Comp. Soc. Press, 2003.
33. P. Laud. Encryption cycles and two views of cryptography. In *Proc. 7-th Nordic Workshop on Secure IT Systems (NORDSEC)*, Karlstad Univ. Studies No. 31, pp. 85–100, 2002.
34. P. Laud. Symmetric encryption in automatic analyses for confidentiality against active adversaries. In *Proc. 2004 IEEE Symp. on Security and Privacy*, pp. 71–85, IEEE Comp. Soc. Press, 2004.
35. P. Laud and R. Corin. Sound computational interpretation of formal encryption with composed keys. In *Proc. 6-th International Conf. on Information Security and Cryptology (ICISC)*, Springer LNCS Vol. 2971, pp. 55–66, 2003.

36. P. Lincoln, J. C. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic polynomial-time framework for protocol analysis. In *Proc. 5-th ACM Conf. on Computer and Communications Security (CCS)*, pp. 112–121, ACM Press, 1998.
37. S. Micali and P. Rogaway. Secure computation. In *CRYPTO '91*, Springer LNCS Vol. 576, pp. 392–404, 1991.
38. S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. *SIAM J. Computing*, 17(2):412–426, 1998.
39. D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Proc. 2-nd Theory of Cryptography Conference (TCC 2005)*, Springer LNCS Vol. 3378, pp. 169–187, 2005.
40. D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *J. Computer Security*, 12(1):99–130, 2004. Prelim. version in WITS'02.
41. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proc. 1-st Theory of Cryptography Conference (TCC 2004)*, Springer LNCS Vol. 2951, pp. 133–151, 2004.
42. 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 on `http://theory.stanford.edu/people/jcm/publications.htm`. Prelim. report in FOSSACS'04, Springer LNCS Vol. 2987.
43. B. Pfitzmann, M. Schunter, and M. Waidner. Cryptographic security of reactive systems. *DERA/RHUL Workshop on Secure Architectures and Information Flow*, 1999, ENTCS, 2000. http://www.elsevier.nl/cas/tree/store/tcs/free/noncas/pc/menu.htm.
44. B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In *Proc. 7-th ACM Conf. on Computer and Communications Security*, pp. 245–254, ACM Press, 2000. Extended version (with M. Schunter) IBM Research Report RZ 3206, 2000, http://www.zurich.ibm.com/security/models.
45. B. Pfitzmann and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 2001 IEEE Symp. on Security and Privacy*, pp. 184–200, IEEE Comp. Soc. Press, 2001.
46. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40-th IEEE Symp. on Foundations of Comp. Sci. (FOCS)*, pp. 543–553, IEEE Comp. Soc. Press, 1999.
47. B. Warinschi. A computational analysis of the Needham-Schröeder-(Lowe) protocol. In *Proc. 16-th IEEE Computer Security Foundations Workshop (CSFW)*, pp. 248–262, IEEE Comp. Soc. Press, 2003.

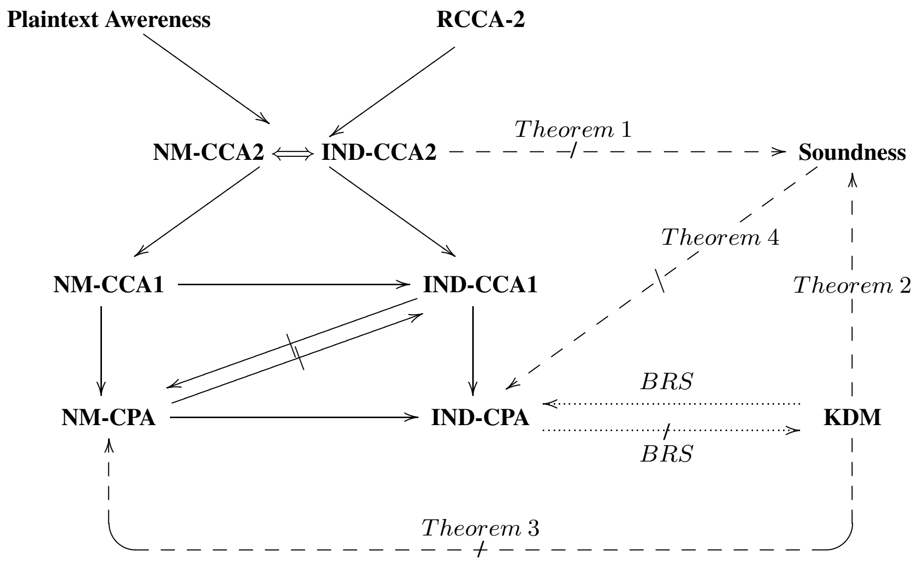
A Computational Definitions of Security For Asymmetric Encryption Schemes

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

B Interpretation Algorithm

For a pattern M we define the interpretation as

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



(Contributions of this paper are represented by dashed arrows.)

Fig. 1. Relation Among Different Security Notions

algorithm $\mathrm{CONVERT}_\eta(M)$
    if $M = K$ where $K \in$ **Keys then**
        return $\tau(K)$
    if $M = K^{-1}$ where $K \in$ **Keys**$^{-1}$ **then**
        return $\tau(K^{-1})$
    if $M = B$ where $B \in$ **Blocks then**
        return B
    if $M = (M_1, M_2)$ **then**
        $x \longleftarrow \mathrm{CONVET}_\eta(M_1)$
        $y \longleftarrow \mathrm{CONVERT}_\eta(M_2)$
        return $[x, y]$
    if $M = \{M_1\}_K$ **then**
        $x \longleftarrow \mathrm{CONVERT}_\eta(M_1)$
        $y \longleftarrow \mathcal{E}(\tau(K), x)$
        return y
    if $M = \square_{K,\ell(M')}$, **then**
        $y \longleftarrow \mathcal{E}(\tau(K), 0^{|\Phi_\eta(M')|})$
        return y

We note that expressions are simply patterns in which symbols of the form $\square_{K,\ell(M')}$ do not appear, and thus can be interpreted by this same algorithm.