Two Views of Cryptography and the Gap In-Between

by

Zehou Wu
B.A., University of British Columbia, 2023

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm/Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Two Views of Cryptography and the Gap In-Between

by

Zehou Wu
B.A., University of British Columbia, 2023

Supervisory Committee

Dr. Bruce Kapron, Supervisor
(Department of Computer Science)

Dr. Yun Lu, Supervisor
(Department of Computer Science)

ABSTRACT

There are two popular views of cryptography. One is formal (symbolic), which uses expressions to model the ideal functionality of encryption functions and is easy to verify. The other is computational, which is what cryptographic assumptions rely on and is used for most security definitions. The challenge of reconciling these two views of cryptography lies in security under the presence of encryption cycles.

In this thesis, we provide a proof of completeness for Abadi-Rogaway symbolic logic with respect to KDM security, a strong form of circular security. Further, we provide a larger set of expressions for which Micciancio's symbolic logic is complete with respect to CPA security, extending Micciancio's completeness, which holds only for the set of acyclic expressions. We also give an alternate characterization of Micciancio's logic. On the computational side, we give a proof that circular insecurity is maintained as cycle length decreases, which is not a previously shown result.

Contents

ACKNOWLEDGEMENTS

I thank Dr. **Bruce Kapron** for admitting me into this Master's program. He gave me a shot at academia, I will go beyond his expectations.

It has been my pleasure to have **Dr. Bruce Kapron** and **Dr. Yun Lu** as my supervisors. They introduced me to cryptography, differential privacy, and many academic processes.

I also thank **Dr. Mohammad Hajiabadi** for taking me as a Ph.D. student. I will cherish this opportunity.

Lastly, I want to thank all parties that have supported me financially, this includes **my parents**, **Dr. Bruce Kapron** and **Dr. Yun Lu**.

DEDICATION

To anyone who has taken the time to read this thesis.
After all, what is a paper without an audience?

Chapter 1

Introduction

Cryptography is a branch of mathematics, science, and engineering often associated with security and communication. However, as theoretical computer scientists, we understand cryptography as the study of (intended) inabilities of functions or Turing machines.

This view was captured by two classic approaches, symbolically or computationally, each with its benefits. The symbolic approach defines formal expressions over atomic sets of keys and data to represent an idealized encryption functionality and adversarial knowledge is modeled via an entailment relation over expressions [1, 15, 19, 20, 26, 32, 33, 38, 40, 46–48]. This approach has the advantage of being more easily applied to formal verification. [10, 18, 21, 37]. The computational approach defines algorithms over bit-strings, and adversarial knowledge is modeled by the output of any algorithm in a class of Turing machines (typically probabilistic and polynomial time) [11, 24, 52]. Most modern cryptographic definitions and primitives are defined in such a manner [31].

Abadi and Rogaway sought to unify these two views [2]; they showed that, in the absence of encryption cycles, equivalence between expressions in the symbolic setting implies indistinguishability in the computational setting. This is described as soundness, where the symbolic side is considered a language, and the computational interpretation is its semantics. The dual of soundness is completeness, where if two cipher texts are computationally indistinguishable, then their corresponding expressions should be symbolically equivalent. Their reasoning for such restriction (the absence of encryption cycles) was justified since, at the time, it was thought best to avoid encryption cycles in practice. As time passed, this was no longer the case.

An encryption cycle consists of a sequence of keys, each encrypting the next,

forming a cycle. We say an encryption scheme is circular secure if it remains secure under the presence of encryption cycles. At the time of [2], Abadi and Rogaway considered only security against Chosen Plaintext Attack (CPA), where the adversary may choose the underlying plaintexts of the ciphertext that it wants to distinguish. Such a definition of security was not designed to face key dependent messages (such as encryption cycles) in any way. Security under Key Dependent Message (KDM) was formally defined a few years later [9], which requires security even when messages may arbitrarily depend on the keys. This motivated more research on circular security. Circular security has become significant in the construction of various primitives whose security relies on underlying primitives with circular security properties. For example, LWE-based encryption with an additional circular security assumption is used to construct homomorphic encryption [14, 23] and indistinguishability obfuscation [22]. A well-known example of how circular encryption arises in practice is Windows BitLocker [45], where a key may be used to encrypt a disk that the key is stored on as data.

1.1 Background and Motivation

With the goal of showing soundness and completeness between the formal expressions and computational interpretations, many works studied how encryption cycles can be handled on the formal side. [4,17,28,36,41–44]. In 2009, Adão, Bana, Herzog and Scedrov showed, along with other results, that Abadi-Rogaway logic is sound with respect to KDM security [4], removing the need to restrict attention to acyclic expressions. A year later, Micciancio, using co-induction to define symbolic adversarial knowledge, showed soundness with respect to CPA security, for arbitrary expressions [41]. This is referred to as Micciancio's logic. Given these general soundness results, it is natural to ask whether there are corresponding completeness results. More recently, Micciancio proved completeness in the absence of encryption cycles [42], but was not able to extend completeness beyond this class of expressions. In this paper, he asserts that acyclicity is a natural condition for establishing completeness with respect to CPA. Our results give a counter-example to this assertion for limited expressions.

However, any constructive proof for completeness involves finding encryption schemes that are circular insecure. There have been numerous works dealing with the (im)possibility of circular security under different scenarios, [3, 5–8, 12, 13, 16, 25, 27, 29, 34, 35, 39, 49]. In terms of circular security, it was shown in [51] that KDM

security, a more demanding notion than that of circular security [9], is reducible to 1-circular bit security. Further, it was shown in [12] that a circular secure encryption scheme can be created using the DDH assumption. However, a counter-example to a stronger security notion may not be a counter-example to a weaker security notion. Although there exists folklore construction on circular insecurity of length 1 cycle (see Appendix A.2), for cycles of longer lengths, the constructions rely on specific assumptions, such as SXDH, indistinguishability obfuscation, or LWE [5, 8, 16, 25, 35, 49].

These previous works led to two questions:

1. Is Abadi-Rogaway logic complete with respect to KDM security without assumptions on acyclicity?
2. Is Micciancio's logic complete with respect CPA security?

We answer the first question with an *yes*. This, combined with [4], implies that Abadi-Rogaway logic is sound and complete with respect to KDM security. However, the second question is not so easily answered. Micciancio showed this logic is complete for acyclic expressions with respect to CPA security [41]. We expand on this set of expressions and show that Micciancio's logic is complete for expressions that may contain length 1 encryption cycles. We also provide the following two results that take a step towards proving the completeness of Micciancio's logic in the future. The first is an alternative characterization of *equivalence* in Micciancio's logic. The second is a reduction from any larger circular counter-example to a smaller one. An alternative characterization may help find a proof of completeness by showing that the computational indistinguishability in the CPA setting shares the same characteristics. However, in any constructive proof of completeness, there is a need to construct a circular counter-example. Our reduction from a longer cycle to shorter cycles comes to use when desiring a circular insecure counter-example of many different lengths (see Appendix A.1 for an example).

1.2 Our Results

We prove that Abadi-Rogaway logic is complete with respect to Key Dependent Message security by showing that if two expressions are not symbolically equivalent, then their computational interpretations using some KDM secure encryption scheme are distinguishable. We also extend Micciancio's proof of completeness for his logic with respect to CPA security, in particular giving a restricted class of cyclic expressions

for which it holds. These results are shown in the symmetric key setting. Finally, we present two related results that may shed some light on our goal of proving completeness for general expressions. The first is an alternate approach to Micciancio's logic. In particular, we show that the equivalence characterized by this logic is the smallest equivalence relation that satisfies two simple and natural properties. Then, turning our attention to the computational setting, we show that a CPA-secure encryption scheme that is n-circular insecure implies the existence of such a scheme that is n'-circular insecure. This last result is shown in the public key setting but applies to the symmetric key setting as well.

1.3 Organization

We provide a relevant definition of formal security in Chapter 2. In Chapter 3, we provide computational definitions of security where subsection 3.1 focuses on symmetric key setting. Subsection 3.2 explained what computational interpretation is. Subsection 3.3 focuses on public key setting. We show Abadi-Rogaway logic is complete with respect to KDM in Chapter 4. We show the completeness of Micciancio's logic with respect to length 1 key cycles in Chapter 5. We characterize the equivalence relation of Micciancio's logic in Chapter 6. Finally, we show that the existence of n-circular insecure encryption scheme implies n'-circular insecure encryption scheme for $n' < n$ in Chapter 7.

Chapter 2

Symbolic Security

We work in the framework for symbolic cryptography introduced by Abadi and Rogaway [2], which focuses on characterizing equivalence between formal cryptographic expressions in a way that is faithful to computational interpretations. We largely follow the presentation given by Micciancio in [41].

In this approach, the basic syntactic classes are data and keys. By $Data$ we denote the set of *data* elements, and by Key the set of *keys*. *Expressions* are defined as follows

$$Exp ::= Data \mid Key \mid (Exp, Exp) \mid \{\!|Exp|\!\}_{Key}$$

We will use $E, E', E_0, E_1, E_2, \ldots$ to denote expressions, $K, K', K_0, K_1, K_2, \ldots$ to denote elements of Key and $B, B', B_1, B_2, \ldots$ to denote elements of $Data$. (E_1, E_2) is a pair of expressions, and $\{\!|E|\!\}_K$ represents encrypting E under key K.

Patterns are obtained by adding unique terms $\Box$ and $\circ$ to $Data$ and Key respectively. These are intended to represent data or keys that are not recoverable.

$$\begin{aligned} Pattern ::= Data \cup \{\Box\} \mid Key \cup \{\circ\} \mid (Pattern, Pattern) \\ \mid \{\!|Pattern|\!\}_{Key \cup \{\circ\}} \end{aligned}$$

Note that every expression is a pattern. The *shape* of a pattern E, denoted $shape(E)$ is the pattern obtained by replacing every element of $Data$ in E by $\Box$ and every element of Key by $\circ$. For example the shape of the pattern $\{\!|K_1, B_1|\!\}_{K_2}$ is $\{\!|\circ, \Box|\!\}_{\circ}$. This will be used later to describe "indistinguishability"; for example, the expression $\{\!|E|\!\}_K$ appears to be "equivalent" to $\{\!|\Box|\!\}_K$ to an adversary without knowledge of key K.

We define keys and parts of a pattern as follows:

$$\begin{aligned}
\mathbf{Keys}(B) &= \emptyset \\
\mathbf{Keys}(K) &= \{K\} \\
\mathbf{Keys}((E_1, E_2)) &= \mathbf{Keys}(E_1) \cup \mathbf{Keys}(E_2) \\
\mathbf{Keys}(\{\!|E|\!\}_K) &= \{K\} \cup \mathbf{Keys}(E)
\end{aligned}$$

$$\begin{aligned}
\mathbf{Parts}(B) &= \{B\} \\
\mathbf{Parts}(K) &= \{K\} \\
\mathbf{Parts}((E_1, E_2)) &= \mathbf{Parts}(E_1) \cup \mathbf{Parts}(E_2) \\
\mathbf{Parts}(\{\!|E|\!\}_K) &= \{\{\!|E|\!\}_K\} \cup \mathbf{Parts}(E)
\end{aligned}$$

We use $r(E)$ to denote $\mathbf{Keys}(E) \cap \mathbf{Parts}(E)$. We also define the pattern function $\mathbf{p} : \mathit{Pattern} \times \mathit{Keys} \rightarrow \mathit{Pattern}$.

$$\begin{aligned}
\mathbf{p}(B, T) &= B \\
\mathbf{p}(K, T) &= K \\
\mathbf{p}((E_1, E_2), T) &= (\mathbf{p}(E_1, T), \mathbf{p}(E_2, T)) \\
\mathbf{p}(\{\!|E|\!\}_K, T) &= \begin{cases} \{\!|\mathit{shape}(E)|\!\}_K & \text{If } K \notin T \\ \{\!|\mathbf{p}((E, T)|\!\}_K & \text{otherwise.} \end{cases}
\end{aligned}$$

The function $\mathbf{p}$ can be thought of as how an expression is viewed when the available knowledge is the set of keys T.

The operators $\mathbf{p}$ and r satisfy the following three properties [41].

1. $\mathbf{p}((E, \mathbf{Keys}) = E$.
2. $\mathbf{p}((\mathbf{p}((E, S), T) = \mathbf{p}(E, S \cap T)$.
3. $r(\mathbf{p}(E, T)) \subseteq r(E)$.

Using the concepts described so far, we define the following operator which captures the fundamental task of *key recovery* with respect to a set of known keys. For a set A, we use $P(A)$ to denote the power set of A.

Definition 1 (Key recovery operators)**.** *For an expresison E the operator $\mathcal{F}_E : P(Key) \to P(Key)$ is defined as $\mathcal{F}_E(T) = r(\boldsymbol{p}(E, T))$.*

This function is monotone and has both least and greatest fixed points (see, e.g., [41, Theorem 1].) We write $\mathrm{fix}(\mathcal{F}_E)$ to denote the least fixed point and $\mathrm{FIX}(\mathcal{F}_E)$ to denote the greatest fixed point. The first of these fixed points models an *inductive adversary* and corresponds to the keys that any adversary *must* know on seeing E, while the latter models a *co-inductive adversary* and corresponds to the keys that any adversary *might* know on seeing E (e.g. subject to prior knowledge.)

In the system of Abadi and Rogaway, two expression E_1 and E_2 are *symbolically equivalent up to renaming* if

$$\mathbf{p}(E_1, \mathrm{fix}(\mathcal{F}_{E_1})) = \sigma\mathbf{p}(E_2, \mathrm{fix}(\mathcal{F}_{E_2}))$$

where σ is a key renaming bijection. In Micciancio's logic, two expressions E_1 and E_2 are symbolically equivalent up to renaming if

$$\mathbf{p}(E_1, \mathrm{FIX}(\mathcal{F}_{E_1})) = \sigma\mathbf{p}(E_2, \mathrm{FIX}(\mathcal{F}_{E_2})).$$

We will use $\mathrm{Pattern}(E)$ to denote $\mathbf{p}(E, \mathrm{FIX}(\mathcal{F}_E))$ and $\mathrm{pattern}(E)$ to denote $\mathbf{p}(E, \mathrm{fix}(\mathcal{F}_E))$

We say E_1, E_2 are *immediate sub-patterns of* (E_1, E_2) and E is an *immediate sub-pattern of* $\{E\}_K$. The *sub-pattern* relation is the reflexive transitive closure of the immediate sub-pattern relation. Similar relations may be defined for expressions.

For patterns E_1 and E_2 we write $E_1 \cong E_2$ if there exists a key renaming bijection σ such that $E_1 = \sigma E_2$. Thus, Micciancio's notion of symbolic equivalence can be characterized as

$$\mathrm{Pattern}(E) \cong \mathrm{Pattern}(F),$$

and Abadi-Rogaway's notion of symbolic equivalence can be characterized as

$$\mathrm{pattern}(E) \cong \mathrm{pattern}(F).$$

In these cases we write $E \sim_{\mathrm{FIX}} F$ and $E \sim_{\mathrm{fix}} F$ respectively.

Definition 2. *The operator $\mathcal{F}_E^i(S)$ is defined as follows:*

- $\mathcal{F}_E^1(S) = \mathcal{F}_E(S)$
- $\mathcal{F}_E^{i+1}(S) = \mathcal{F}_E(\mathcal{F}_E^i(S))$.

It is a well-known property of fixed points that

$$\mathrm{FIX}(\mathcal{F}_E) = \mathcal{F}_E^{|E|}(Key)^1$$

and

$$\mathrm{fix}(\mathcal{F}_E) = \mathcal{F}_E^{|E|}(\emptyset).$$

[1] $|E|$ is the number of Data symbols and Key symbols in E including repetitions

Chapter 3

Computational Semantics

In this chapter, we provide computational security definitions. In Section 3.1, we define symmetric encryption schemes and their relevant security definitions. This includes security under Chosen Plaintext Attack (CPA), Left-Right Oracle (LR), Authentication (AE), and Key-Dependent Message (KDM). These definitions are used in Chapter 4 and 5. We provide how expressions are interpreted with computational semantics in Section 3.2. In Section 3.3, we define public key encryption schemes and their relevant security definitions. This includes security under Chosen Plaintext Attack (CPA) and n-Circular in Section 3.3. These definitions apply to Chapter 7.

For a string x, we write $|x|$ to denote the length of the string. For a randomized function f with input x, we write $y \leftarrow f(x)$ to indicate that y is the output of $f(x)$ with fresh randomness. A negligible function $\mathrm{negl}(\cdot)$ is a function such that for every positive integer c, there exists an integer N_c, if $x \geq N_c$, then $\mathrm{negl}(x) < \frac{1}{x^c}$.

3.1 Symmetric-key Encryption Schemes

Definition 3. *A* symmetric key encryption scheme *consists of three efficiently computable randomized functions: a key generating function $\mathcal{G}$, encryption function $\mathcal{E}$ and decryption function $\mathcal{D}$. These operate on a key space, message space and ciphertext space, each of which is some subset of $\{0,1\}^*$.*

- *The key generating function $\mathcal{G}$ takes a unary string 1^η, where η is the security parameter and uniformly outputs an element from the key space.*
- *The encryption function $\mathcal{E}$ takes a key k and a message m and outputs a cipher text c.*

- *The decryption function $\mathcal{D}$ takes a key k and a ciphertext c and outputs a message m or $\perp$.*

As usual, we require that for any key k and message m, $\mathcal{D}(k, \mathcal{E}(k, m)) = m$. Without loss of generality, we will assume that there is a fixed polynomial p such that, on input 1^η, the key generation algorithm returns an element of $\{0,1\}^{p(\eta)}$.

The definition of chosen plain text attack (CPA) experiment and CPA security follows that of [31].

Definition 4. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a symmetric key encryption scheme. The CPA indistinguishability experiment for Π and adversary $\mathcal{A}$ is denoted by $\mathsf{CPA}_{\mathcal{A},\Pi}(\eta)$ and defined as follows:*

$\mathsf{CPA}_{\mathcal{A},\Pi}(\eta)$:

1. $k \leftarrow \mathcal{G}(1^\eta)$, $b \leftarrow_R \{0,1\}$
2. $\mathcal{A}$ *is given* 1^η *and oracle access to* $\mathcal{E}(k,\cdot)$ *and outputs two equal length messages* m_0, m_1
3. $\mathcal{A}$ *receives* $c \leftarrow \mathcal{E}(k, m_b)$.
4. $\mathcal{A}$ *continues to have access to* $\mathcal{E}(k,\cdot)$ *and outputs* $b' \in \{0,1\}$
5. *If* $b = b'$ *return* 1 *else return* 0.

A symmetric encryption scheme Π is CPA-secure if for any PPT adversary $\mathcal{A}$

$$\Pr[\mathsf{CPA}(\mathcal{A}, \Pi, \eta) = 1] \leq \frac{1}{2} + negl(\eta)$$

where η is the security parameter.

An experiment equivalent to the CPA experiment is the left-right-oracle (LR-oracle) experiment.

Definition 5. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a symmetric key encryption scheme. Let the oracle $\hat{\mathcal{E}}$ be defined by $\hat{\mathcal{E}}(k, b, m_0, m_1) = \mathcal{E}(k, m_b)$ if $|m_0| = |m_1|$ and $\perp$ otherwise. The LR-oracle experiment for Π and adversary $\mathcal{A}$ is denoted by $\mathsf{LR}_{\mathcal{A},\Pi}(\eta)$ and defined as follows:*

$\mathsf{LR}_{\mathcal{A},\Pi}(\eta)$

1. $k \leftarrow \mathcal{G}(1^\eta)$, $b \leftarrow_R \{0,1\}$

2. *$\mathcal{A}$ is given 1^η and oracle access to $\hat{\mathcal{E}}(k, b, \cdot, \cdot)$ and outputs $b' \in \{0,1\}$.*

3. *If $b = b'$ return 1 else return 0.*

Later, we will also need to consider an extended notion of security, namely authenticated encryption. The following definition of the Authenticated Encryption (AE) experiment follows that given in [16].

Definition 6. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a symmetric key encryption scheme. Let oracles $\hat{\mathcal{E}}$ and $\hat{\mathcal{D}}$ be defined by*

$$\hat{\mathcal{E}}(k, b, m_0, m_1) = \begin{cases} \mathcal{E}(k, m_b) & \text{if } |m_0| = |m_1| \\ \perp & \text{otherwise} \end{cases}$$

and

$$\hat{\mathcal{D}}(k, m) = \begin{cases} \mathcal{D}(k, m) & \text{if the query is not previous output of } \hat{\mathcal{E}} \\ \perp & \text{otherwise} \end{cases}$$

respectively. The AE experiment is denoted by $\mathsf{AE}_{\mathcal{A},\Pi}(\eta)$ and defined as follows:

$\mathsf{AE}_{\mathcal{A},\Pi}(\eta)$

1. *$k \leftarrow \mathcal{G}(1^\eta)$, $b \leftarrow_R \{0,1\}$.*

2. *$\mathcal{A}$ is given 1^η and oracle access to $\hat{\mathcal{E}}(k, b, \cdot, \cdot), \hat{\mathcal{D}}(k, \cdot)$ and outputs $b' \in \{0,1\}$.*

3. *If $b = b'$ return 1 else return 0.*

A symmetric encryption scheme Π is AE-secure or is an AE if for any PPT adversary $\mathcal{A}$,

$$\Pr[\mathsf{AE}(\mathcal{A}, \Pi, \eta) = 1] \leq \frac{1}{2} + \mathit{negl}(\eta)$$

where η is the security parameter.

We note that while AE appears to be a stronger notion than CPA, it too can be obtained from the minimal assumption of the existence of one-way functions (see, e.g., [16].)

The following definition of the key dependent message (KDM) experiment follows the original definition of KDM given in [9].

Definition 7. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a symmetric key encryption scheme. Let $n = n(\eta)$ be some non-zero polynomial of η. Let the encryption oracle $\hat{\mathcal{E}}$ be defined by*

$$\hat{\mathcal{E}}^{K,b}(i, g) = \begin{cases} \mathcal{E}(k_i, g(K)) & \text{if } b = 1; \\ \mathcal{E}(k_i, 0^{|g(K)|}) & \text{otherwise.} \end{cases}$$

The KDM experiment for Π and adversary $\mathcal{A}$, denoted by $\mathsf{KDM}(\mathcal{A}, \Pi, \eta)$ is defined as follows:

1. *$K \leftarrow (k_1, \ldots, k_n)$ where $k_i \leftarrow \mathcal{G}(1^\eta)$ for $1 \leq i \leq n$, $b \leftarrow_R \{0, 1\}$.*
2. *$\mathcal{A}$ is given 1^η and oracle access to $\hat{\mathcal{E}}^{K,b}(i, \cdot)$ and outputs $b' \in \{0, 1\}$.*
3. *If $b = b'$ return 1 else return 0.*

A symmetric encryption scheme Π is KDM-secure if for any PPT adversary $\mathcal{A}$

$$\Pr[\mathsf{KDM}(\mathcal{A}, \Pi, \eta) = 1] \leq \frac{1}{2} + negl(\eta)$$

where η is the security parameter.

While KDM security implies CPA security, as discussed above the status of the converse implication remains open.

3.2 Computational Interpretation of Expressions

This subsection contains how expressions are interpreted in the computational setting. We also provide definitions of soundness and completeness between a logic and a class of encryption schemes.

The computational evaluation/interpretation of an expression E with respect to a scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ with security parameter η, denoted $[\![E]\!]_\Pi$, relies on the following:

- a length regular function γ that maps each data block B to a string b.
- an efficient pairing function $(\cdot, \cdot)$.
- a key mapping τ that assigns to each unique K_i a freshly generated $k_i \leftarrow \mathcal{G}(1^\eta)$.

We assume γ is fixed, and for a given Π define $[\![E]\!]_\Pi$ inductively by

1. $[\![B_i]\!]_\Pi = \gamma(B_i)$
2. $[\![K_i]\!]_\Pi = \tau(K_i)$
3. $[\![(E_1, E_2)]\!]_\Pi = ([\![E_1]\!]_\Pi, [\![E_2]\!]_\Pi)$
4. $[\![\{\!|E_1|\!\}_K]\!]_\Pi = \mathcal{E}([\![K]\!]_\Pi, [\![E_1]\!]_\Pi)$, each call to $\mathcal{E}$ uses independent randomness.

In cases where we need to refer to a specific assignment τ, we write $\tau[\![E]\!]_\Pi$.

In [42, Lemma 8], Micciancio gives an encoding in which expressions of different shapes are mapped to strings of different lengths. This is a modification of the computational interpretation given above. We restate his Lemma here

Lemma 8. *For any encryption scheme* Π *and any two expressions* E_0 *and* E_1 *with* $shape(E_0) \neq shape(E_1)$*, there exists a computational interpretation such that*

$$|e_0| \neq |e_1|$$

where $e_0 \leftarrow [\![E_0]\!]_\Pi$ *and* $e_1 \leftarrow [\![E_1]\!]_\Pi$.

We assume that our computational interpretation $[\![\cdot]\!]$ satisfies this property.

Definition 9. *Let* $\mathcal{C}$ *be a class of symmetric key encryption schemes and* E_0, E_1 *expressions. We say that* E_0 *and* E_1 *are* $\mathcal{C}$-equivalent*, denoted* $E_0 \approx_\mathcal{C} E_1$ *if for every* $\Pi \in \mathcal{C}$, $[\![E_0]\!]_\Pi$ *is computationally indistinguishable from* $[\![E_1]\!]_\Pi$.

Definition 10. *Let* $\mathcal{S}$ *be a symbolic security model (e.g.* $\mathcal{S} \in \{fix, FIX\}$*) and* $\mathcal{C}$ *a class of symmetric key encryption schemes. We say that* $\mathcal{S}$ *is* sound with respect to $\mathcal{C}$ *if for all expressions* E_0, E_1*,*

$$E_0 \sim_\mathcal{S} E_1 \Longrightarrow E_0 \approx_\mathcal{C} E_1.$$

Likewise, $\mathcal{S}$ *is* complete with respect to $\mathcal{C}$ *if for all expressions* E_0, E_1*,*

$$E_0 \approx_\mathcal{C} E_1 \Longrightarrow E_0 \sim_\mathcal{S} E_1.$$

We will continue to use $\mathcal{C}$ to denote the class of symmetric key encryption schemes for the rest of this work.

Note that it is often easier to reason about the contrapositive version of completeness, namely, for all expressions E_0, E_1,

$$E_0 \not\sim_{\mathcal{S}} E_1 \implies \exists \Pi \in \mathcal{C}([\![E_0]\!]_\Pi \not\approx [\![E_1]\!]_\Pi)$$

We note that this formulation implicitly relies on the assumption that $\mathcal{C} \neq \emptyset$. We will leave this assumption implicit, as completeness holds trivially in the case where the assumption fails.

Some prior works define completeness differently, namely, for all expressions E_0, E_1,

$$\forall \Pi \in \mathcal{C}([\![E_0]\!]_\Pi \approx [\![E_1]\!]_\Pi \implies E_0 \sim_{\mathcal{S}} E_1).$$

We will call this notion *strong completeness*. The notion we use is that of [42]. This is a weaker notion of completeness, but it corresponds to the typical notion completeness for a deductive system namely, if a formula φ is valid (i.e. satisfiable in *all* models), then it has a derivation in the deductive system. It is also more suited to reasoning about general notions of security, as opposed to individual encryption schemes. On the other hand, completeness immediately has a nontrivial cryptographic consequence. In particular, as long as there exists some E, E' such that $E \not\sim_{\mathcal{S}} E'$, then the completeness of $\mathcal{S}$ with respect to $\mathcal{C}$ implies that $\mathcal{C}$ is nonempty. Given this unavoidable consequence, an important question of interest is whether or not nonemptiness of $\mathcal{C}$ is an adequate condition to guarantee completeness.

3.3 Public-key Encryption Schemes

Definition 11. *A* public key encryption scheme *consists of three efficiently computable randomized functions: a key generating function $\mathcal{G}$, encryption function $\mathcal{E}$, and decryption function $\mathcal{D}$.*

- *The key generating function $\mathcal{G}$ takes a unary string 1^η, where η is the security parameter and outputs a pair, $(\mathsf{pk}, \mathsf{sk})$, consisting of a public key and a secret key. We assume for any $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathcal{G}(1^\eta)$ and $(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathcal{G}(1^\eta)$ that $|\mathsf{pk}_i| = |\mathsf{pk}_j|$ and $|\mathsf{sk}_i| = |\mathsf{sk}_j|$.*

- *The encryption function $\mathcal{E}$ takes a public key pk and a message m and outputs a cipher text c.*

- *The decryption function $\mathcal{D}$ takes a secret key sk and a ciphertext c and outputs a message m or $\perp$.*

As usual, we require that for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}(1^\eta)$ and any message m, $\mathcal{D}(\mathsf{sk}, \mathcal{E}(\mathsf{pk}, m)) = m$ except with negligible probaiblity.

A (public key) bit encryption scheme has message space $\{0,1\}$. For a bit encryption scheme and a message m where $|m| > 1$, we write $\mathcal{E}(\mathsf{pk}, m)$ to denote that bit-by-bit encryption of m using $\mathcal{E}(\mathsf{pk}, \cdot)$, each encryption using fresh randomness. More definitions of bit-encryption in circular settings can be found [49].

The definition of CPA experiment and CPA-security follows that of [31].

Definition 12. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. The CPA experiment for Π and adversary $\mathcal{A}$ denoted by $\mathsf{CPA}_{\mathcal{A},\Pi}(\eta)$ is defined as follows:*

$\mathsf{CPA}_{\mathcal{A},\Pi}(\eta)$:

1. $\mathsf{pk}, \mathsf{sk} \leftarrow \mathcal{G}(1^\eta)$. *Give η and pk to $\mathcal{A}$.*
2. $b \leftarrow \$\,\{0,1\}$.
3. *$\mathcal{A}$ outputs two equal length challenge messages m_0, m_1.*
4. *Compute $c^b \leftarrow \mathcal{E}(k, m_b)$ and return c^b to $\mathcal{A}$.*
5. *$\mathcal{A}$ outputs b' and the experiment results in 1 if $b = b'$, 0 otherwise.*

We say Π is CPA secure if for every PPT adversary $\mathcal{A}$,

$$\Pr[\mathsf{CPA}_{\mathcal{A},\Pi}(\eta) = 1] \leq \frac{1}{2} + negl(\eta).$$

We can observe that in the public key setting, the adversary does not need access to an encryption oracle since it possesses the public key.

The definitions of the n-circular experiment and n-circular-security follow that of [16].

Definition 13. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. The n-circular experiment for Π and adversary $\mathcal{A}$ denoted by $\mathsf{CIRC}_{\mathcal{A},\Pi}(n, \eta)$ is defined as follows:*

$\mathsf{CIRC}_{\mathcal{A},\Pi}(n, \eta)$:

1. $\mathsf{pk}_0, \mathsf{sk}_0 \leftarrow \mathcal{G}(1^\eta), \ldots, \mathsf{pk}_{n-1}, \mathsf{sk}_{n-1} \leftarrow \mathcal{G}(1^\eta)$. *Give η and $\mathsf{pk}_0, \ldots, \mathsf{pk}_{n-1}$ to $\mathcal{A}$.*

2. $b \leftarrow \$\,\{0,1\}$.

3. *Compute for* $i \in \{0,\ldots,n-1\}$*:*

$$c_i^b \leftarrow \begin{cases} \mathcal{E}(\mathsf{pk}_i, \mathsf{sk}_{i+1 \bmod n}) & \textit{If } b = 1 \\ \mathcal{E}(\mathsf{pk}_i, 0^{|\mathsf{sk}_1|}) & \textit{If } b = 0 \end{cases}$$

Send $c_0^b, , \ldots, c_{n-1}^b$ *to* $\mathcal{A}$*.*

4. *When* $\mathcal{A}$ *outputs* b'*, the experiment results in* 1 *if* $b = b'$*,* 0 *otherwise.*

We say Π *is* n*-circular secure if for every PPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathit{CIRC}_{\mathcal{A},\Pi}(n,\eta) = 1] \leq \frac{1}{2} + \mathit{negl}(\eta).$$

Definition 14 (Circular Insecurity)**.** *An encryption scheme is* n*-circular insecure if it is not* n*-circular secure. An encryption scheme is (1 to* n*)-circular insecure if it is* ℓ*-circular insecure for any* $\ell \in \{1,\ldots,n\}$*.*

Although it is unknown whether CPA security implies circular security or not, CPA security provides indistinguishability on ciphertexts that are "almost" an encryption cycle. What we mean by this is the following: consider a sequence of ciphertexts that form an encryption cycle, if we replace some of these ciphertexts with encryptions of zeros, then this sequence is indistinguishable from the sequence of ciphertexts that are all encryptions of zeros with the same keys. We denote this experiment between an adversary $\mathcal{A}$ and an encryption scheme Π as $\mathsf{ZERO}_{\mathcal{A},\Pi}(n,t,\eta)$. It is required that $n, t \geq 1$, and η is the security parameter. The parameter n is the length of an encryption cycle or just the number of ciphertexts in a sequence when encrypting zeros. In the experiment, the first t ciphertexts of an encryption cycle are replaced with encryptions of zero.

$\mathsf{ZERO}_{\mathcal{A},\Pi}(n,t,\eta)$:

1. $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}(1^\eta), \ldots, (\mathsf{pk}_{n-1}, \mathsf{sk}_{n-1}) \leftarrow \mathcal{G}(1^\eta)$. Send η and $\mathsf{pk}_0, \ldots, \mathsf{pk}_{n-1}$ to $\mathcal{A}$.

2. A random coin b is flipped.

3. For $i \in \{0, \ldots, n-1\}$ compute

$$c_i^b \leftarrow \begin{cases} \mathcal{E}(\mathsf{pk}_i, \mathsf{sk}_{i+1 \bmod n}) & \text{if } b = 1 \text{ and } i \geq t, \\ \mathcal{E}(\mathsf{pk}_i, 0^{|\mathsf{sk}_1|}) & \text{otherwise.} \end{cases}$$

   Send $c_0^b, \ldots, c_{n-1}^b$ to $\mathcal{A}$.

4. When $\mathcal{A}$ outputs b', the experiment result in 1 if $b = b'$, 0 otherwise.

Lemma 15. *If a public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is CPA secure. Then for any PPT adversary $\mathcal{A}$, any $n \geq 1$, and any $t \geq 1$,*

$$\Pr[\mathit{ZERO}_{\mathcal{A},\Pi}(n, t, \eta) = 1] \leq \frac{1}{2} + \mathit{negl}(\eta)$$

for all but finitely many η and some negligible function negl.

Applying the soundness result from [41] is an easy method to show this. However, we provide the following proof to keep this thesis self-contained without elaborating on formal cryptography.

Proof. Assume that Π is CPA secure. We will show that for any PPT adversary $\mathcal{A}$, and some negligible function negl,

$$|\Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t, \eta) = 1] - \Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t+1, \eta) = 1]| \leq \mathrm{negl}(\eta)$$

for any $n, t \geq 1$. The statement then follows, since n is not dependent on η and for $t' \geq n$ it must be the case that

$$\Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t', \eta) = 1] = \frac{1}{2}.$$

This is because when $t' \geq n$, the adversary is receiving encryptions of zeros in the experiment regardless of the value of b.

Consider an arbitrary adversary $\mathcal{A}$ such that

$$|\Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t, \eta) = 1] - \Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t+1, \eta) = 1]| = \frac{1}{2} + \epsilon(\eta).$$

Now consider an adversary $\mathcal{A}'$ to play in $\mathsf{CPA}_{\mathcal{A}',\Pi}(\eta)$ defined as follows.

$\mathcal{A}'$:

1. Receive η and pk. Label this pk as pk_t.
2. For $i \in \{0, \ldots, n-1\} \backslash \{t\}$ compute $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathcal{G}(1^\eta)$. Give $\mathsf{pk}_0, \ldots, \mathsf{pk}_{n-1}$ to $\mathcal{A}$.
3. Query challenge messages $m_0 = 0^{|\mathsf{sk}|}$ and $m_1 = \mathsf{sk}_{t+1}$ to receive $c_t^b \leftarrow \mathcal{E}(\mathsf{pk}_t, m_b)$.
4. For $i \in \{0, \ldots, n-1\} \setminus \{t\}$ compute

$$c_i \leftarrow \begin{cases} \mathcal{E}(\mathsf{pk}_i, 0^{|\mathsf{sk}|}) & \text{if } i < t \\ \mathcal{E}(\mathsf{pk}_i, \mathsf{sk}_{i+1 \bmod n}) & \text{otherwise.} \end{cases}$$

   Give $c_0, \ldots, c_t^b, \ldots, c_n$ to $\mathcal{A}$.
5. When $\mathcal{A}$ outputs b', output b'.

It can be observed that when $b = 0$, $\mathcal{A}'$ has simulated $\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t+1, \eta)$ and when $b = 1$, $\mathcal{A}'$ has simulated $\mathsf{ZERO}_{\mathcal{A},\Pi}(n, t, \eta)$. Therefore, we conclude that if $\epsilon(\eta)$ is not negligible, then Π is not CPA secure. $\square$

Chapter 4

Completeness of Inductive Equivalence with Respect to KDM

We prove that KDM is complete for inductive reasoning via a more general result, in particular showing that it is complete for any class of encryption schemes closed under a construction (inspired by a similar technique from [42]) which we now define.

Construction 1. *Given a symmetric key encryption scheme* $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$*, we obtain* $\Pi_1 = (\mathcal{G}_1, \mathcal{E}_1, \mathcal{D}_1)$ *by the* key revealing construction *as follows:*

- $\mathcal{G}_1(1^\eta)$ *calls* $\mathcal{G}(1^\eta)$ *twice independently creating* k_0 *and* k_1 *and returns* $k = \langle k_0, k_1 \rangle$*.*
- $\mathcal{E}_1(\langle k_0, k_1 \rangle, m)$ *returns* $\langle k_0, \mathcal{E}(k_1, m) \rangle$*.*
- $\mathcal{D}_1(\langle k_0, k_1 \rangle, \langle k_0', c \rangle)$ *returns* $\mathcal{D}(k_1, c)$*.*

Using this construction, we can show the following theorem.

Theorem 1. *If a non-empty class of symmetric key encryption schemes,* $\mathcal{C}$*, is closed under the key revealing construction, then the least fixed point semantics is complete with respect to* $\mathcal{C}$*, i.e.,*

$$E_0 \approx_{fix} E_1 \implies \exists \Pi \in \mathcal{C}, [\![E_0]\!]_\Pi \not\approx [\![E_1]\!]_\Pi.$$

We begin defining the following computational key recovery function, using an approach similar to that of [44].

Algorithm 1 Algorithm $\mathrm{C}_{\mathrm{kr}}(e, T_i)$

if e is a data block **then** return $\emptyset$
else if e is a key **then** return $\{e\}$
else if e is a pair (e_0, e_1) **then** return $\mathrm{C}_{\mathrm{kr}}(e_0, T_i) \cup \mathrm{C}_{\mathrm{kr}}(e_1, T_i)$
else if e is a cipher **then**
    we know now $e = \langle k_0, c \rangle$
    if $\exists \langle k_0', k_1' \rangle \in T_i$ s.t. $k_0' = k_0$ **then** return $\mathrm{C}_{\mathrm{kr}}(\mathcal{D}(k_1', c), T_i)$.
                                ▷ $\mathcal{D}$ is the decryption function
    else return $\emptyset$
    end if
end if

Algorithm 2 Algorithm `Crecoverable`(e)

$T_0 \leftarrow \emptyset$, $i \leftarrow 0$
repeat
    $i \leftarrow i + 1$
    $T_i \leftarrow \mathrm{C}_{\mathrm{kr}}(e, T_{i-1})$
until $T_i = T_{i-1}$ or $i \geq |e|$
return T_i

The key-revealing construction and computational key recovery function are defined in a complementary way to give us exactly what we need for completeness, using the following main ideas:

- KDM security is preserved under the key-revealing construction (Proof of Theorem 2[1])
- Computational key recovery against a key-revealing scheme captures exactly the power of an inductive symbolic key-recovery adversary (Lemma 17)

The following lemma shows that C_{kr} captures $\mathcal{F}_E$.

Lemma 16. *For any expression E, $S \subseteq \boldsymbol{Keys}(E)$ and $\eta > 0$*

$$\Pr\left[C_{kr}(e, \tau[S]) \neq \tau[\mathcal{F}_E(S)]\right] \leq |S||e| negl(\eta),$$

where $e \leftarrow \tau[\![E]\!]_{\Pi_1}$ and for each $k \in \boldsymbol{Keys}(E)$, $\tau(k) \in \{0,1\}^\eta$ is chosen independently at random.

Proof. We show this statement via structural induction on E.

[1]We note that CPA security is also preserved by the construction

- Consider $E = B$. Then $e = b$ and $\mathrm{C}_{\mathrm{kr}}(e, \tau[S]) = \emptyset = \tau[\mathcal{F}_E(S)]$. Therefore

$$Pr\left[\mathrm{C}_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_E(S)]\right] = 0.$$

- Consider $E = K$. Then $e = \tau K$ and $\mathrm{C}_{\mathrm{kr}}(e, \tau[S]) = \tau[\{K\}] = \tau[\mathcal{F}_E(S)]$. Therefore

$$Pr\left[\mathrm{C}_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_E(S)]\right] = 0.$$

Assume the statement holds for expressions E_0 and E_1. Let $e_i \leftarrow \tau[\![E_i]\!]_{\Pi_1}$ for $i \in \{0, 1\}$.

- Consider $E = (E_0, E_1)$. Then $e = (e_0, e_1)$ and

$$\begin{aligned}
&\Pr[\mathrm{C}_{\mathrm{kr}}((e_1, e_2), \tau[S]) \neq \tau[\mathcal{F}_{(E_1,E_2)}(S)]] \\
&\quad \leq \Pr[\mathrm{C}_{\mathrm{kr}}(e_1, \tau[S]) \neq \tau[\mathcal{F}_{E_1}(S)]] + \Pr[\mathrm{C}_{\mathrm{kr}}(e_2, \tau[S]) \neq \tau[\mathcal{F}_{E_2}(S)]] \\
&\quad \leq |S||e_1|\mathrm{negl}(\eta) + |S||e_2|\mathrm{negl}(\eta) && \text{(by IH)} \\
&\quad \leq |S||(e_1, e_2)|\mathrm{negl}(\eta).
\end{aligned}$$

- Consider $E = \{\!|E_0|\!\}_K$. Then $e = \mathcal{E}_1(\tau K, e_0) = \langle k_0, \mathcal{E}(\tau k_1, e_0)\rangle$ where $\tau K = \langle k_0, k_1\rangle$. Then either $K \in S$ or $K \notin S$.
  Case 1. If it is the case that $K \in S$, then τK must be in $\tau[S]$. Since every key in $\tau[S]$ is generated with fresh randomness, the probability that any key symbol that is not K is evaluated to $\langle k_0, x\rangle$ is $\leq |S|\mathrm{negl}(\eta)$. Therefore we have

$$\begin{aligned}
&\Pr[\mathrm{C}_{\mathrm{kr}}(\langle \tau K[0], \mathcal{E}(\tau K[1], e_0)\rangle, \tau[S]) \neq \tau[\mathcal{F}_{\{E_0\}_K}(S)]] \\
&\quad \leq 1 - (\Pr[\mathrm{C}_{\mathrm{kr}}(\langle \tau K[0], \mathcal{E}(\tau K[1], e_0)\rangle, \tau[S]) = \tau[\mathcal{F}_{\{\!|E_0|\!\}_K}(S)]]) \\
&\quad \leq 1 - ((1 - |S|\mathrm{negl}(\eta))(1 - |S||e_0|\mathrm{negl}(\eta))) && \text{(by IH)} \\
&\quad \leq |S||e|\mathrm{negl}(\eta).
\end{aligned}$$

  Case 2. Consider $K \notin S$. The probability that $\langle k_0, x\rangle \in \tau[S]$ is $\leq |S|\mathrm{negl}(\eta)$ Therefore,

$$\begin{aligned}
&\Pr[\mathrm{C}_{\mathrm{kr}}(\langle \tau K[0], \mathcal{E}(\tau K[1], e_0)\rangle, \tau[S]) \neq \tau[\mathcal{F}_{\{\!|E_0|\!\}_K}(S)]] \\
&\quad \leq |S|\mathrm{negl}(\eta).
\end{aligned}$$

□

Now we provide the following lemma that shows `Crecoverable` corresponds to the least fixed point.

Lemma 17. *For any expression E and $\eta > 0$*

$$\Pr[\textit{Crecoverable}(e) \neq \tau[\mathit{fix}(\mathcal{F}_E)]] \leq |e|^3 \mathit{negl}(\eta),$$

where $e \leftarrow \tau[\![E]\!]_{\Pi_1}$.

Proof. We will use induction on i to show

$$\Pr[T_i \neq \tau[\mathcal{F}_E^i]] \leq i|e|^2\mathrm{negl}(\eta).$$

Base Case. Consider $i = 1$. We have $T_1 = \mathrm{C}_{\mathrm{kr}}(e, \emptyset)$ and $\mathcal{F}_E^1(\emptyset) = \mathcal{F}_E(\emptyset)$. By Lemma 16, we have that

$$\Pr[T_1 \neq \tau[\mathcal{F}_E(\emptyset)]] \leq |e|\mathrm{negl}(\eta).$$

Assume the statement holds for i. Consider $i + 1$. We have $T_{i+1} = \mathrm{C}_{\mathrm{kr}}(e, T_i)$ and $\mathcal{F}_E^{i+1}(\emptyset) = \mathcal{F}_E(\mathcal{F}_E^i(\emptyset))$. By induction hypothesis we have

$$\Pr[T_i \neq \tau[\mathcal{F}_E^i(\emptyset)] \leq i|e|^2\mathrm{negl}(\eta).$$

Therefore

$$\begin{aligned}
&\Pr[\mathrm{C}_{\mathrm{kr}}(e, T_i) \neq \mathcal{F}_E^{i+1}(\emptyset)] \\
&= 1 - \Pr[\mathrm{C}_{\mathrm{kr}}(e, T_i) = \mathcal{F}_E^{i+1}(\emptyset)] \\
&= 1 - \Pr[\mathrm{C}_{\mathrm{kr}}(e, \tau[\mathcal{F}_E^i(\emptyset)]) = \tau[\mathcal{F}_E(\mathcal{F}_E^i(\emptyset))]] \mid T_i = \tau[\mathcal{F}_E^i(\emptyset)]] \cdot \Pr[T_i = \tau[\mathcal{F}_E^i]] \\
&= 1 - (1 - \Pr[\mathrm{C}_{\mathrm{kr}}(e, \tau[\mathcal{F}_E^i(\emptyset)]) \neq \tau[\mathcal{F}_E(\mathcal{F}_E^i(\emptyset))]] \mid T_i = \tau[\mathcal{F}_E^i(\emptyset)]]) \cdot \Pr[T_i = \tau[\mathcal{F}_E^i(\emptyset)]] \\
&\leq 1 - (1 - |e|^2\mathrm{negl}(\eta)) \cdot (1 - i|e|^2\mathrm{negl}(\eta)) \\
&\leq (i + 1)|e|^2\mathrm{negl}(\eta)
\end{aligned}$$

□

Since $\texttt{Crecoverable}(e) = T_i$ for some $i \leq |e|$, we have

$$\Pr[\texttt{Crecoverable}(e) \neq \tau[\text{fix}(\mathcal{F}_E)]] \leq |e|^3 \text{negl}(\eta).$$

Proof of Theorem 1. Recall that by Lemma 8, expressions of different shape will be evaluated to strings of different lengths. We proceed to consider only the cases where the shape of the expressions are equal but either some data blocks differ or some keys differ after renaming.

1. $\mathit{shape}(E_0) = \mathit{shape}(E_1)$ but data blocks in corresponding positions of $\mathbf{p}(E_1, \text{fix}(\mathcal{F}_{E_1}))$ and $\mathbf{p}(E_2, \text{fix}(\mathcal{F}_{E_2}))$ differ. Let $T(E_i)$, $i = 0, 1$ denote the list of all data blocks in $p(E_i, \text{fix}(\mathcal{F}_{E_i}))$ ordered as they appear from left to right. By assumption there exists an index j where $T(E_1)[j] \neq T(E_2)[j]$. Now when the adversary gets $e_b \leftarrow [\![E_b]\!]_{\Pi_1}$ for $b \leftarrow \{0, 1\}$ it uses $\texttt{Crecoverable}'(e_b)$ to recover a set of keys which with all but negligible probability correspond to $\tau[\text{fix}(\mathcal{F}_{E_b})]]$, and uses these to produce a list of strings corresponding to recoverable data blocks, ordered as they appeared left to right in e_b. Finally, it returns b' such that the jth recoverable data block strings corresponds to jth data block of $T(E_{b'})$.

2. $\mathit{shape}(E_0) = \mathit{shape}(E_1)$ and all data block in the patterns are equal, but some keys are not equal after renaming. Define the function $r(\cdot, \cdot, \cdot)$ that takes as parameters an expression E, and key position p and p':

$$r(E, p, p') = \begin{cases} 1 & \text{If for } E, \text{ the keys at position } p \text{ and } p' \text{ are equal.} \\ 0 & \text{Otherwise.} \end{cases}$$

   In this case, there must exist position p and p' such that $r(\text{Pattern}(E_0), p, p') \neq r(\text{Pattern}(E_1), p, p')$, (recall that $\text{Pattern}(E) = \mathbf{p}(E, \text{fix}(\mathcal{F}_E))$). As in the previous case, using Lemma 17 the adversary can check all pairs of key positions in e_b to determine b with all but negligible probability.

□

Now, to specifically show that Abadi-Rogaway logic is complete with respect to KDM secure encryption schemes, we show KDM security is preserved under Construction 1.

Theorem 2. *If the class of KDM secure symmetric encryption schemes is not empty, then*

$$E_0 \nsim_{fix} E_1 \implies \exists \Pi \in KDM, [\![E_0]\!]_\Pi \not\approx [\![E_1]\!]_\Pi.$$

Proof. It suffices to show that the key-revealing construction preserves KDM security. Consider an arbitrary KDM-secure encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ and let $\Pi_1 = (\mathcal{G}_1, \mathcal{E}_1, \mathcal{D}_1)$ be obtained from Π via the key-revealing construction. Consider an arbitrary PPT adversary $\mathcal{A}_1$ where

$$\Pr[\mathsf{KDM}(\mathcal{A}_1, \Pi_1, \eta) = 1] \geq \frac{1}{2} + \epsilon(\eta).$$

We construct a KDM adversary $\mathcal{A}$ against Π which works by simulating the KDM game for Π_1, using $\mathcal{A}_1$ as subroutine.

$\mathcal{A}(1^\eta)$:

1. Send 1^η to $\mathcal{A}_1$.
2. Receive the key vector length n. Call $\mathcal{G}(1^\eta)$ n times to produce keys $k'_1, \ldots, k'_n$, let $\overline{K'} = k'_1, \ldots, k'_n$, and send n to $\mathcal{A}_1$.
3. Whenever $\mathcal{A}_1$ makes a query (j, g) for index j and circuit g, create a circuit g' as follows:
   - g' takes as input a key vector $\overline{K}$
   - g' creates a key vector $\overline{K; K'}$ where the i-th element is $\langle k'_i, k_i \rangle$ and runs g on $\overline{K; K'}$.

   Query the KDM oracle at (j, g') to receive output c, and send $\langle k'_j, c \rangle$ to $\mathcal{A}_1$.
4. If $\mathcal{A}_1$ outputs b', output b'.

One can observe that $\mathcal{A}$ runs in polynomial time with respect to η if $\mathcal{A}_1$ runs in polynomial time with respect to η.

Each of the n keys in the KDM game for $\mathcal{A}$ is generated by calling $\mathcal{G}(1^\eta)$ as are the keys, $k'_1, \ldots, k'_n$. This means that each $\langle k'_i, k_i \rangle$ is identical to an output of $\mathcal{G}_1(1^\eta)$. Therefore $\overline{K'; K}$ is distributed identically to what $\mathcal{A}_1$ expects in $\mathsf{KDM}(\mathcal{A}_1, \Pi_1, \eta)$.

In the case where $b = 1$ in $\mathsf{KDM}(\mathcal{A}, \Pi, \eta)$, if the query received from $\mathcal{A}_1$ is (j, g), then $\mathcal{A}$'s response to $\mathcal{A}_1$ is

$$k'_j, \mathcal{E}(k_j, g'(\overline{K}))).$$

Since $g'(\overline{K}) = g(\overline{K', K})$, this is identical to response when querying (j, g) in $\mathsf{KDM}(\mathcal{A}_1, \Pi_1, \eta)$ when its internal coin is 1.

In the case where $b = 0$ in $\mathsf{KDM}(\mathcal{A}, \Pi, \eta)$, if the query from $\mathcal{A}_1$ is (j, g), then $\mathcal{A}$'s response is

$$(k'_j, \mathcal{E}(k_j, 0^{g'(\overline{K})})).$$

Since the output length of g is fixed and that the output of g' and g have the same length, therefore $0^{g'(\overline{K})} = 0^{g(\overline{K';K})}$. Hence this is identical to the response when querying (j, g) in $\mathsf{KDM}(\mathcal{A}_1, \Pi_1, \eta)$ when its internal coin is 0.

Therefore we have

$$\begin{aligned} \Pr[\mathsf{KDM}(\mathcal{A}, \Pi, \eta) = 1] &= \Pr[\mathsf{KDM}(\mathcal{A}_1, \Pi_1, \eta) = 1] \\ &\geq \frac{1}{2} + \epsilon(\eta). \end{aligned}$$

In conclusion, if Π is KDM secure, then so is Π_1. □

In Theorem 2, we showed that Abadi-Rogaway logic is complete with respect to KDM security. Combining this result with the soundness result from [4], we have both soundness and completeness for Abadi-Rogayway logic with respect to KDM security. In other words, the knowledge of a symbolic adversary in Abadi-Rogaway logic, represented by the least fixed point of the key recovery function, captures that of an adversary upon seeing a ciphertext from a KDM-secure encryption scheme. However, it can be argued that the greatest fixed point of the key recovery operator better represents an adversary's knowledge in the CPA setting. We investigate this in the next section.

Chapter 5

CPA-Completeness of Coinduction for Restricted Classes of Expressions

In [42], Micciancio proves coinductive reasoning is complete for CPA, in a more general model allowing pseudorandom keys. However, he does so only for a class of *acyclic expressions*. Here acyclic is used in a strong syntactic sense (originating in [2]): an expression E is acyclic if and only if it contains no subexpressions of the form $\{E'\}_K$, where K has any occurrence in E'. The question of whether such completeness holds for broader class of expressions is left open, and it is noted that "least fixed points are the best fit for completeness proofs". We conjecture that CPA-completeness does in fact hold for coinductive adversaries, and towards establishing this conjecture we will prove completeness for a class of expressions which extend the acyclic ones considered in [42].

An immediate question that arises is whether the CPA-completeness of coinduction has any cryptographic implications, beyond the obvious requirement noted above, namely the existence of a CPA-secure encryption scheme, and hence of one-way functions.

If we consider any class of expressions including cyclic expressions of the form $(\{\!|K_1|\!\}_{K_2}, \ldots, \{\!|K_n|\!\}_{K_1})$, completeness will imply the existence of CPA-secure schemes that are not n-circular secure, even with respect to adversaries that do not have access to an encryption oracle. The existence of such schemes is currently an unsettled problem without additional assumptions. We note that [16] prove, based only on

the existence of one-way functions, that there exist CPA-secure schemes that are not "weakly n-circular secure". However, the existence of these schemes does not imply the existence of the above-mentioned schemes implied by completeness.

An immediate extension of completeness beyond the acyclic expressions considered by [42] is to broaden the class of acyclic expressions. In particular, we will say that an expression is *semantically acyclic* if $\mathrm{FIX}(\mathcal{F}_E) = \mathrm{fix}(\mathcal{F}_E)$. Not every semantically acyclic expression is syntactically acyclic, e.g., $\{\!|\{\!|K_1|\!\}_{K_2}|\!\}_{K_1}$. For this class of expressions, CPA-completeness follows immediately from Theorem 1.

5.1 CPA-Completeness of Coinduction for Restricted Classes of Expressions

As a step towards our conjecture, we will prove completeness for a simple class of expressions, namely the *syntactically 1-circular* expressions defined below in Subsection 5.1.1. Our proof relies on the folklore construction of a CPA-secure encryption scheme that is not 1-circular secure (based on the original observation of [24].) In the setting where the adversary has access to a CPA encryption oracle, any n-cycle is also insecure.

To begin, to accommodate the extended class of expressions under consideration, we need to give an extended key revealing scheme and key recovery function. As the definition of the scheme uses key recovery, we define the key recovery functions first:

Algorithm 3 Algorithm $\mathrm{C}'_{\mathrm{kr}}(e, T)$

if $e = b$ is a data block **then**
    return $\emptyset$
else if $e = k$ is a key **then**
    return $\{k\}$
else if $e = (e_0, e_1)$ is a pair **then**
    return $\mathrm{C}'_{\mathrm{kr}}(e_0, T) \cup \mathrm{C}'_{\mathrm{kr}}(e_1, T)$
else if $e = \langle b, k, x, c\rangle$ is a ciphertext, where $b \in \{0, 1\}$ and $|x| = |k|$ **then**
    if $\langle k, k_i\rangle$ is in T for some key value k_i **then**
        return $\mathrm{C}'_{\mathrm{kr}}(\mathcal{D}_1(\langle k, k_i\rangle, e), T)$ ▷ $\mathcal{D}_1$ is the decryption function
    else if $e = \langle 1, k_0, k_1, c\rangle$ is a ciphertext **then**
        return $\{\langle k_0, k_1\rangle\}$.
    else return $\emptyset$
    end if
else
    return $\emptyset$.
end if

Algorithm 4 Algorithm `Crecoverable′`(e)

$T_0 \leftarrow \emptyset$, *cont* $\leftarrow 1$.
repeat
    $i \leftarrow i + 1$
    $T_i \leftarrow \mathrm{C}'_{\mathrm{kr}}(e, T_{i-1})$
until $T_i = T_{i-1}$ or $i \geq |e|$
return T_i

One can observe that for any expression $\mathrm{C}'_{\mathrm{kr}}$ and `Crecoverable′` both run in polynomial time with respect to length of e, therefore if the length of e is polynomial with respect to η (which is certainly the case when e is the interpretation of an expression E with respect to any reasonable encryption function,) then so is the runtime of `Crecoverable′`

The purpose of this computational key recovering algorithm to recover the keys needed to distinguish syntactic 1-circular expressions with respect to the class of schemes defined in Construction 2. The corresponding correctness is shown in Lemma 23.

Construction 2. *Given a symmetric key encryption scheme* $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$*, we obtain* $\Pi_1 = (\mathcal{G}_1, \mathcal{E}_1, \mathcal{D}_1)$ *as follows:*

- $\mathcal{G}_1(\eta)$: *call* $\mathcal{G}(1^\eta)$ *twice to obtain* k_0 *and* k_1. *Return* $\langle k_0, k_1\rangle$.
- $\mathcal{E}_1(\langle k_0, k_1\rangle, m)$: *Let* $S := \texttt{Crecoverable}'(m)$.
  Return $\begin{cases} \langle 1, k_0, k_1, \mathcal{E}(k_1, m)\rangle & \text{if } (k_0, k_1) \in S; \\ \langle 0, k_0, 0^{|k_1|}, \mathcal{E}(k_1, m)\rangle & \text{otherwise.} \end{cases}$
- $\mathcal{D}_1(\langle k_0, k_1\rangle, \langle b, k_0', x, c\rangle)$: *Return* $\mathcal{D}(k_1, c)$.

The encryption function $\mathcal{E}_1$ makes a call to $\texttt{Crecoverable}'$, and $\texttt{Crecoverable}'$ makes calls to $\mathcal{D}_1$, but there are no mutual calls. A ciphertext created by Π_1 consists of a flag, the first half of a key, the second half of a key, and a smaller ciphertext. The encryption function $\mathcal{E}_1$ given inputs key k and message m uses $\texttt{Crecoverable}'$ to check if k is "contained in" m, and sets the flag and the second half of a key in ciphertext accordingly.

In Section 5.1.1, we will use Π_1 to obtain completeness for the greatest fixed point semantics on a restricted class of formulas with respect to CPA-secure encryption. To begin, we show that Π_1 is CPA secure if Π is AE-secure. The reason we start with an AE-secure encryption scheme is that it has a simple proof of security against an adversary trying to recover the secret key. We justify using an AE-secure encryption scheme because if a CPA-secure encryption scheme exists, then an AE-secure encryption scheme must also exist. This is because both of these notions are equivalent to the existence of a one-way function [30].

Below, we present a concise proof that it is hard for an adversary to recover a secret key against an AE-secure encryption scheme.

Lemma 18. *If the scheme* Π *is AE secure, then for any PPT adversary* $\mathcal{A}_1$

$$\Pr_{k \leftarrow \mathcal{G}(1^\eta)}[\mathcal{A}_1^{\mathcal{E}(k,\cdot)}(\eta) = k] \leq \mathit{negl}(\eta)$$

Proof. Consider an arbitrary efficient adversary $\mathcal{A}_1$ such that

$$\Pr_{k \leftarrow \mathcal{G}(1^\eta)}[\mathcal{A}_1^{\mathcal{E}(k,\cdot)}(\eta) = k] = \epsilon(\eta).$$

We can construct an adversary $\mathcal{A}$ to play $\mathsf{AE}(\mathcal{A}, \Pi, \eta)$ as follows:

$\mathcal{A}$:

1. Receive η.

2. Let $Q := \emptyset$.
3. Send η to $\mathcal{A}_1$.
4. When $\mathcal{A}_1$ queries message m, generate encryption query (m, m) to receive $c \leftarrow \mathcal{E}(k, m)$ and return c to $\mathcal{A}$. Let $Q := Q \cup \{m\}$.
5. When $\mathcal{A}_1$ outputs k', choose an $m \notin Q$ and compute $c_{k'} \leftarrow \mathcal{E}(k', m)$.
6. Query the decryption oracle on $c_{k'}$ to receive m'. Return 1 if $m' \neq \perp$, return a random bit otherwise.

Since the secret key k of $\mathsf{AE}(\mathcal{A}, \Pi, \eta)$ is generated using $\mathcal{G}(1^\eta)$, the interaction between $\mathcal{A}$ and $\mathcal{A}_1$ is identical to the interaction between the encryption oracle $\mathcal{E}(k, \cdot)$ and $\mathcal{A}_1$. Therefore we have that $\Pr[k' = k] = \epsilon(\eta)$.

In the event $b = 1$, either $k' = k$ or not.

- If $k' = k$ then $m' = m$ by the correctness of Π and therefore $\mathcal{A}$ outputs 1.
- Otherwise, if $m' = \perp$ then $\mathcal{A}$ outputs 1 with probability $1/2$, and otherwise $\mathcal{A}$ outputs 1 with probability 1. So $\mathcal{A}$ outputs 1 with probability $\geq 1/2$.

In the event $b = 0$, $m' = \perp$ with probability 1 and hence $\mathcal{A}$ outputs 0 with probability $1/2$.

Hence we have

$$\begin{aligned}
&\Pr[\mathsf{AE}(\mathcal{A}, \Pi, \eta) = 1] \\
&\quad \geq \frac{1}{2}\left(\epsilon(\eta) + (1 - \epsilon(\eta))\frac{1}{2}\right) + \frac{1}{2}\left(\frac{1}{2}\right) \\
&\quad = \frac{1}{2} + \frac{\epsilon(\eta)}{4}.
\end{aligned}$$

□

Lemma 19. *The scheme Π_1 from Construction 2 is CPA-secure, assuming Π is AE-secure.*

Proof. We show this via a sequence of games using an "identical-until-bad" argument. Assume Π is AE-secure and fix an efficient adversary $\mathcal{A}_1$.

Game 0. *Let Game 0 be* $\mathsf{LR}(\mathcal{A}_1, \Pi_1, \eta)$*, we state the encryptions explicitly with respect to* Π *and add a "set flag" operation in line 4.*

1. $k' \leftarrow \mathcal{G}(1^\eta), k \leftarrow \mathcal{G}(1^\eta)$.
2. *A random coin* b *is flipped.*
3. *Give* b *to* $\mathcal{A}_1$.
4. *Whenever* $\mathcal{A}_1$ *queries messages* (m_0, m_1)*, computes* $c_b \leftarrow \mathcal{E}(k, m_b)$.
   If $\langle k', k\rangle \in \texttt{Crecoverable}'(c)$*, set "bad" and return* $\langle 1, k', k, c_b\rangle$ *to* $\mathcal{A}_1$*. Else return* $\langle 0, k', 0^k, c_b\rangle$ *to* $\mathcal{A}_1$.
5. *When* $\mathcal{A}_1$ *output* b' *the experiment results in* 1 *if* $b = b'$*,* 0 *otherwise.*

We let S_0 *denote the event where Game 0 results in* 1.

Game 1. *Let Game 1 is Game 0, with line 4 modified:*

1. $k' \leftarrow \mathcal{G}(1^\eta), k \leftarrow \mathcal{G}(1^\eta)$.
2. *A random coin* b *is flipped.*
3. *Give* b *to* $\mathcal{A}_1$.
4. *Whenever* $\mathcal{A}_1$ *queries messages* (m_0, m_1)*, computes* $c_b \leftarrow \mathcal{E}(k, m_b)$*. Return* $\langle 0, k', 0^k, c_b\rangle$ *to* $\mathcal{A}_1$.
5. *When* $\mathcal{A}_1$ *output* b' *the experiment results in* 1 *if* $b = b'$*,* 0 *otherwise.*

Let S_1 *denote the event where Game* 1 *results in 1.*

It can be observed that if "bad" is never set in Game 0 then it is identical to Game 1. Therefore we have

$$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[\text{"bad" is set in Game 0}].$$

By Lemma 18, we have the following inequality:

$$\begin{aligned}&\Pr[\text{"bad" is set in Game 0}] \geq \Pr[\langle k', k\rangle \in \texttt{Crecoverable}'[\{m \mid \mathcal{A} \text{ queried } m\}]] \\ &\geq \text{negl}(\eta) \cdot |\texttt{Crecoverable}'[\{m \mid \mathcal{A} \text{ queried } m\}]|\,.\end{aligned}$$

Since the $\mathcal{A}_1$ is efficient $|\texttt{Crecoverable}'[\{m \mid \mathcal{A} \text{ queried } m\}]|$ must be a polynomial of η, and therefore $\Pr[\text{“bad” is set in Game 0}]$ is a negligible function of η. Now we show that $\Pr[S_0] \leq \frac{1}{2} + \text{negl}(\eta)$ by building the following adversary $\mathcal{A}$ to play in $\mathsf{AE}(\mathcal{A}, \Pi, \eta)$.

$\mathcal{A}$**:** 1. Receive η, compute $k' \leftarrow \mathcal{G}(1^\eta)$ and send it to $\mathcal{A}_1$.

2. Whenever $\mathcal{A}$ queries (m_0, m_1), make encryption query (m_0, m_1) to receive $c_b \leftarrow \mathcal{E}(k, m_b)$. Return $\langle 0, k', 0^{|k|}, c_b \rangle$ to $\mathcal{A}_1$.

3. When $\mathcal{A}_1$ outputs b' output b'.

Since $\mathcal{A}$ simulates Game 1 perfectly for $\mathcal{A}_1$, by AE-security of Π, we have

$$\Pr[\mathsf{AE}(\Pi, \mathcal{A}, \eta) = 1] = \Pr[S_1] = \Pr[S_0] + \text{negl}(\eta).$$

We conclude that Π_1 is CPA-secure. □

Below we will use Π_1 in conjunction with the modified computational key recovery function to prove completeness for a class of acyclic expressions. However, we might be concerned that we needed to start with an AE scheme to build Π_1. As discussed above, we would like a result of the form: “if CPA-secure schemes exist at all, then coinduction is complete for CPA-secure schemes”. So far, with the given construction we appear to have something like: “if AE-secure schemes exist at all, then coinduction is complete for CPA-secure schemes”. While this seems like a mismatch, we note that the existence of both AE-secure schemes and CPA-secure schemes is equivalent to the existence of one-way functions. So our construction starting from an AE-secure scheme indeed gives us a result of the form we are looking for.

5.1.1 Syntactically 1-Circular Expressions

An expressions E is *syntactically 1-circular* if the expression E' resulting from replacing every subexpression of the form $\{K_i\}_{K_i}$ by K_i in E is syntactically acyclic.

Example 20. *Consider the following positive and negative examples for syntactic 1-circular expressions.*

- *The expression* $\{\!|K|\!\}_K$ *is syntactically 1-circular.*
- *The expression* $\{\!|(\{\!|K_1|\!\}_{K_1}, \{\!|K_2|\!\}_{K_2})|\!\}_{K_3}$ *is syntactically 1-circular.*

- *The expression* $\{\!|\{\!|K_1|\!\}_{K_1}|\!\}_{K_1}$ *is not syntactically 1-circular.*
- *The expression* $\{\!|(K_1, K_2)|\!\}_{K_1}$ *is not syntactically 1-circular.*

In the rest of this subsection, we show completeness of the greatest fixed point semantics with respect to CPA secure schemes for this set of expressions.

Recall that for a monotone operator $\mathcal{F}_E$, S is a *pre-fixed point of* $\mathcal{F}_E$ if $\mathcal{F}_E(S) \subseteq S$.

Lemma 21. *Let E be syntactically 1-circular and E' be the expression obtained from E by replacing every subexpression of the form $\{\!|K_i|\!\}_{K_i}$ in E by K_i. If S is a pre-fixed point for both $\mathcal{F}_E$ and $\mathcal{F}_{E'}$, then $\mathcal{F}_E(S) = \mathcal{F}_{E'}(S)$*

Proof. We will use structural induction on E.

- **Base Case.** Consider $E = B$. Then $E' = B$ and $\mathcal{F}_E(S) = \emptyset = \mathcal{F}_{E'}(S)$.
- **Base Case.** Consider $E = K$. Then $E' = K$ and $\mathcal{F}_E(S) = \{\!|K|\!\} = \mathcal{F}_{E'}(S)$.

Assume the statement holds for syntactically 1-circular expression E_1 and E_2 and the respective E_1' and E_2'.

- Consider $E = (E_1, E_2)$. Then $E' = (E_1', E_2')$. Since S is a pre-fixed point of $\mathcal{F}_E$ and $\mathcal{F}_{E'}$, we know

$$\mathcal{F}_E(S) = \mathcal{F}_{E_1}(S) \cup \mathcal{F}_{E_2}(S) \subseteq S$$
$$\mathcal{F}_{E'}(S) = \mathcal{F}_{E_1'}(S) \cup \mathcal{F}_{E_2'}(S) \subseteq S$$

  Therefore S is also a pre-fixed point of $\mathcal{F}_{E_1}, \mathcal{F}_{E_2}$ and $\mathcal{F}_{E_1'}, \mathcal{F}_{E_2'}$, so that

$$\begin{aligned} \mathcal{F}_E(S) &= \mathcal{F}_{E_1}(S) \cup \mathcal{F}_{E_2}(S) \\ &= \mathcal{F}_{E_1'}(S) \cup \mathcal{F}_{E_2'}(S) && \text{(by IH)} \\ &= \mathcal{F}_{E'}(S). \end{aligned}$$

- Consider $E = \{\!|E_1|\!\}_K$. Then either $E_1 = K$ or not.

  Case 1. If $E_1 = K$, we have $E' = K$. Since S is pre-fixed point of $\mathcal{F}_{E'}$, $\mathcal{F}_{E'}(S) = \{K\} \subseteq S$, and therefore

$$\mathcal{F}_E(S) = \{K\} = \mathcal{F}_{E'}(S).$$

Case 2. If $E_1 \neq K$, then $E' = \{\!|E'_1|\!\}_K$. Either $K \in S$ or not.

Case 2.1. Assume $K \in S$. Then $\mathcal{F}_E(S) = \mathcal{F}_{E_1}(S)$ and $\mathcal{F}_{E'_1}(S) = \mathcal{F}_{E'_1}(S)$. Therefore S is a pre-fixed point of $\mathcal{F}_{E_1}$ and $\mathcal{F}_{E'_1}(S)$, so that

$$\begin{aligned} \mathcal{F}_E(S) &= \mathcal{F}_{E_1}(S) \\ &= \mathcal{F}_{E'_1}(S) && \text{(by IH)} \\ &= \mathcal{F}_{E'}(S). \end{aligned}$$

Case 2.2. Assume $K \notin S$. Then $\mathcal{F}_E(S) = \emptyset = \mathcal{F}_{E'}(S)$.

□

After replacing every $\{\!|K_i|\!\}_{K_i}$ in E to obtain E', the following property is true.

Lemma 22. *Letting E be a syntactically 1-circular expression and E' is the corresponding expression described above,*

$$\mathit{FIX}(\mathcal{F}_E) = \mathit{fix}(\mathcal{F}_{E'}).$$

Proof. By assumption E' is a syntacticially acyclic, so by [41, Theorem 2], $\mathrm{fix}(\mathcal{F}_{E'}) = \mathrm{FIX}(\mathcal{F}_{E'})$. We now proceed by induction on i to show that $\mathcal{F}^i_E(\mathbf{Keys}) = \mathcal{F}^i_{E'}(\mathbf{Keys})$, to conclude that $\mathrm{FIX}(\mathcal{F}_E) = \mathrm{fix}(\mathcal{F}_{E'})$.

Base Case. Clearly **Keys** is a pre-fixed point of both $\mathcal{F}_E$ and $\mathcal{F}_{E'}$, so by Lemma 21, $\mathcal{F}_E(\mathbf{Keys}) = \mathcal{F}_{E'}(\mathbf{Keys})$.

Assume $\mathcal{F}^i_E(\mathbf{Keys}) = \mathcal{F}^i_{E'}(\mathbf{Keys})$. By the monotonicity of $\mathcal{F}_E$ and $\mathcal{F}_{E'}$ and the fact that $\mathcal{F}^i_E(\mathbf{Keys}) \subseteq \mathbf{Keys}$ we have that

$$\mathcal{F}^{i+1}_E(\mathbf{Keys}) \subseteq \mathcal{F}^i_E(\mathbf{Keys}) \text{ and } \mathcal{F}^{i+1}_{E'}(\mathbf{Keys}) \subseteq \mathcal{F}^i_{E'}(\mathbf{Keys}).$$

By the induction hypothesis $\mathcal{F}^i_E(\mathbf{Keys}) = \mathcal{F}^i_{E'}(\mathbf{Keys})$. Therefore applying Lemma 21 we have

$$\mathcal{F}^{i+1}_E(\mathbf{Keys}) = \mathcal{F}^{i+1}_{E'}(\mathbf{Keys})$$

□

The following lemma shows `Crecoverable'` corresponds to the greatest fixed point for the limited expressions.

Lemma 23. *Let E be an arbitrary syntactically 1-circular expression and E' is the expression where every subexpression of the form $\{\!|K_i|\!\}_{K_i}$ in E is replaced with K_i, then*

$$\Pr[\mathtt{Crecoverable}'(e) \neq \tau[\mathit{fix}(\mathcal{F}_{E'})]] \leq |e|^3 \mathit{negl}(\eta),$$

where $e \leftarrow \tau[\![E]\!]_{\Pi_1}$, Π_1 is the scheme from Construction 2 and η is the security parameter.

Proof. We show this in two parts where first part is similar to Lemma 16 and second part to Lemma 17.

We first show that for any E and any $S \subseteq \mathbf{Keys}(E)$,

$$\Pr[\mathrm{C}'_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_{E'}(S)]] \leq |S||e|\mathrm{negl}(\eta).$$

We will use structural induction on E. The proof of this is very similar to the proof of Lemma 16 with the only difference being the case of $E = \{\!|E_0|\!\}_K$ in the induction step. We show only the case that differs.

Assume the statement holds for syntactic 1-circular expressions E_0. Let E'_0 be E_0 with all subexpression $\{\!|K_j|\!\}_{K_j}$ replaced with K_j for all j and $e_0 \leftarrow \tau[\![E_i]\!]_{\Pi_1}$.

- Consider $E = \{\!|E_0|\!\}_K$. Then $e = \mathcal{E}_1(\tau K, e_0) = \langle b, k_0, x, \mathcal{E}(k_1, e_0)\rangle$ where $\tau K = \langle k_0, k_1\rangle$ and $E' = \{\!|E_0|\!\}_K$. Then either $E_0 = K$ or not.

  Case 1. If $E_1 = K$, then then $e_1 = \tau K$ and $E' = K$. $\mathrm{C}'_{\mathrm{kr}}(e, \tau S) = \tau[\{K\}]$. The probability that a different key symbol is evaluated to $\langle k', x\rangle$ is $\leq |S|\mathrm{negl}(\eta)$

  $$\Pr[\mathrm{C}'_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_{E'}(S)]] \leq |S|\mathrm{negl}(\eta).$$

  Case 2. Assume $E_0 \neq K$. Then either $K \in S$ or not.
  Case 2.1. Assume $K \in S$. Then $\tau K \in \tau[S]$. A different key symbol evaluates to $\langle k_0, x\rangle$ in $\tau[S]$ with probability $\leq |S|\mathrm{negl}(\eta)$. Therefore

  $$\begin{aligned}
  &\Pr[\mathrm{C}'_{\mathrm{kr}}(\langle b, k_0, x, \mathcal{E}(k_1, e_0)\rangle, \tau[S]) \neq \tau[\mathcal{F}_{\{\!|E_0|\!\}_K}(S)]] \\
  &\quad \leq 1 - Pr[\mathrm{C}'_{\mathrm{kr}}(\langle b, k_0, x, \mathcal{E}(k_1, e_0)\rangle, \tau[S]) = \tau[\mathcal{F}_{\{\!|E_0|\!\}_K}(S)]] \\
  &\quad \leq 1 - ((1 - |S|\mathrm{negl}(\eta))(1 - |S||e_0|\mathrm{negl}(\eta))) \qquad \text{(By IH)} \\
  &\quad \leq |S||e|\mathrm{negl}(\eta).
  \end{aligned}$$

Case 2.2. Assume $K \notin S$. Then $\langle k_0, x\rangle \in \tau[S]$ with probability $\leq |S|\mathrm{negl}(\eta)$. Since E' is acyclic, this means that K is not in E_0 and $\mathcal{F}_E(S) = \emptyset$. Hence

$$\begin{aligned}&\Pr[\mathrm{C}'_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_E(S)]]\\ &\leq |S|\mathrm{negl}(\eta).\end{aligned}$$

This concludes

$$\Pr[\mathrm{C}'_{\mathrm{kr}}(e, \tau[S]) \neq \tau[\mathcal{F}_{E'}(S)]] \leq |S||e|\mathrm{negl}(\eta).$$

Using the inequality above and the similar induction as in proof of Lemma 17, we achieve

$$\Pr[\texttt{Crecoverable}'(e) \neq \tau[\mathrm{fix}(\mathcal{F}_{E'}(E))]] \leq |e|^3\mathrm{negl}(\eta).$$

□

Using Lemmas above, we are prepared to show Theorem 3. This states the completeness of Micciancio's logic with respect to CPA security for limited expressions.

Theorem 3. *Coinductive symbolic security is complete with respect to CPA security, for the set of syntactic 1-circular expressions.*

Proof. We will use similar techniques as used in Theorem 1. By Lemma 23 and Lemma 22, for any syntactic 1-circular expression E,

$$\Pr[\texttt{Crecoverable}'(e) = \tau[\mathrm{FIX}(\mathcal{F}_E)]] \geq 1 - |e|^3\mathrm{negl}(\eta).$$

Now techniques from Theorem 1 can be applied to conclude that for any two syntactic 1-circular expressions E_1 and E_1,

$$E_0 \nsim_{\mathrm{fix}} E_1 \implies \exists \Pi \in CPA, [\![E_0]\!]_\Pi \not\approx [\![E_1]\!]_\Pi.$$

□

Chapter 6

An Alternate Characterization of the GFP Semantics

While the results obtained above give some hope for more general forms of completeness, we seem quite far from any concrete approach. On the other hand, it is not really clear how we would go about proving incompleteness. In this section, we give an alternate characterization of the GFP semantics, which might be of help in this direction.

In particular, we show that the equivalence relation given by the greatest fixed point semantics, $\sim_{\text{FIX}}$, is the smallest equivalence relation $\mathcal{R}$ on patterns that satisfies the following two properties.

1. If two patterns E and G are equivalent up to renaming, then $(E, G) \in \mathcal{R}$.
2. If $(\mathbf{p}(E, r(E)), \mathbf{p}(G, r(G))) \in \mathcal{R}$, then $(E, G) \in \mathcal{R}$.

This suggests a possible route to showing incompleteness. Namely, if we can show the existence of *any* equivalence relation satisfying these properties, which is properly contained in CPA-equivalence, we will have incompleteness for coinduction. However, it is not clear that this would be any easier, although we have the leeway to design any equivalence that might work.

We now proceed with showing the alternate characterization. We begin by refining the relation $\sim_{\text{FIX}}$.

Definition 24. *For two patterns E and G, $E \sim_i G$ if*

$$\boldsymbol{p}(E, \mathcal{F}_E^i(\boldsymbol{Keys})) \cong \boldsymbol{p}(G, \mathcal{F}_G^i(\boldsymbol{Keys})).$$

Lemma 25. *If $E \sim_i G$, then $E \sim_{i+1} G$.*

Proof. Assume $E \sim_i G$, which means $\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})) = \sigma\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys}))$ for some key bijection σ. We then have

$$\begin{aligned}
&\mathbf{p}(E,\mathcal{F}_E^{i+1}(\mathbf{Keys})) \\
&\quad = \mathbf{p}(E, r(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})))) \\
&\quad = \mathbf{p}(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})), r(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})))) \\
&\quad = \mathbf{p}(\sigma\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys})), r(\sigma\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys})))) \\
&\quad = \sigma\mathbf{p}(G, r(\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys})))) \\
&\quad = \sigma\mathbf{p}(G, \mathcal{F}_G^{i+1}(\mathbf{Keys}))
\end{aligned}$$

□

Recall that for any monotone operator $\mathcal{F}$, there exists integer j such that $\mathcal{F}_E^j(\mathbf{Keys})) = \mathrm{FIX}(\mathcal{F}_E))$. This gives us the following fact.

Fact 26. $\sim_{FIX} = \bigcup_i \sim_i$.

Theorem 4. *The relation $\sim_{FIX}$ is the smallest equivalence relation $\mathcal{R}$ on patterns that satisfies:*

1. $\cong \subseteq \mathcal{R}$.
2. *If $(\boldsymbol{p}(E, r(E)), \boldsymbol{p}(G, r(G))) \in \mathcal{R}$, then $(E, G) \in \mathcal{R}$.*

Proof. We first show that $\sim_{\mathrm{FIX}}$ satisfies these two properties, then we show it's the smallest equivalence relation satisfying these two properties.

For Property (1), suppose $E \cong G$. Then $E = \sigma G$ for some key bijection σ, so for any i,

$$\mathcal{F}_E^i(\mathbf{Keys}) = \sigma\mathcal{F}_G^i(\mathbf{Keys}),$$

which implies $E \sim_{\mathrm{FIX}} G$. For Property (2), let E and G be patterns such that $\mathbf{p}(E, r(E)) \sim_{\mathrm{FIX}} \mathbf{p}(G, r(G))$. Since $r(E)$ and $r(G)$ are equal to $\mathcal{F}_E^1(\mathbf{Keys})$ and $\mathcal{F}_G^1(\mathbf{Keys})$ respectively, we have that $E \sim_1 G$, so that $E \sim_{\mathrm{FIX}} G$.

To show that $\sim_{\mathrm{FIX}}$ is the smallest equivalence relation satisfying these properties, consider an arbitrary equivalence relation $\mathcal{R}$ that satisfies them. Let E and G be

expressions such that $(E, G) \notin \mathcal{R}$. By Property (1) we know that $E \ncong G$. We use induction on i to show that

$$(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})), \mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys}))) \notin \mathcal{R} \tag{6.1}$$

For the base case we have that

$$\begin{aligned}\mathbf{p}(E,&\mathcal{F}_E^1(\mathbf{Keys})), \mathbf{p}(G, \mathcal{F}_G^1(\mathbf{Keys}))\\ &= \mathbf{p}(E, r(E)), \mathbf{p}(G, r(G)).\end{aligned}$$

So by Property (2), $(\mathbf{p}(E, r(E)), \mathbf{p}(G, r(G))) \notin \mathcal{R}$.

Assume the statement holds for i. For $i+1$, by the induction hypothesis,

$$(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})), \mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys}))) \notin \mathcal{R}.$$

By Property (2), this implies that

$$\begin{aligned}&(\mathbf{p}(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})), r(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})))),\\ &\mathbf{p}(\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys})), r(\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys}))))) \notin \mathcal{R}.\end{aligned}$$

Since

$$\mathbf{p}(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})), r(\mathbf{p}(E, \mathcal{F}_E^i(\mathbf{Keys})))) = \mathbf{p}(E, \mathcal{F}_E^{i+1}(\mathbf{Keys}))$$

and

$$\mathbf{p}(\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys})), r(\mathbf{p}(G, \mathcal{F}_G^i(\mathbf{Keys}))))) = \mathbf{p}(G, \mathcal{F}_G^{i+1}(\mathbf{Keys}))),$$

we have

$$\mathbf{p}(E, \mathcal{F}_E^{i+1}(\mathbf{Keys})), \mathbf{p}(G, \mathcal{F}_G^{i+1}(\mathbf{Keys}))) \notin \mathcal{R},$$

which concludes the induction. Now recall that for any expression G, $\mathrm{FIX}(\mathcal{F}_G) = \mathcal{F}_G^i(\mathbf{Keys})$ for some $i > 0$, so by formula 6.1

$$(\mathbf{p}(E, \mathrm{FIX}(\mathcal{F}_E)), \mathbf{p}(G, \mathrm{FIX}(\mathcal{F}_G))) \notin \mathcal{R}.$$

Since $\mathcal{R}$ satisfies Property (1), this means $\mathrm{Pattern}(E) \ncong \mathrm{Pattern}(G)$, so that $E \nsim_{\mathrm{FIX}} G$. □

Chapter 7

Construction of n-Circular Insecure Encryption

We present new implications for circular (in)security. To be more precise, we show that if an $(n+1)$-circular insecure CPA secure encryption scheme exists, then a $(\leq n)$-circular insecure CPA secure encryption scheme exists. This result applies to both public key and symmetric key encryption schemes. We present the proof for the public key setting, which, with slight modification, can serve as a proof in the symmetric key setting.

The proof uses a circular insecure encryption scheme as a black-box to construct another encryption scheme. In Construction 3, we present a construction that does not preserve the message space of the initial encryption scheme. In Construction 4, we construct an encryption scheme that preserves the message space of the base encryption scheme if the message space of the base scheme consists of only one bit. The techniques used in Construction 4 can be modified to preserve message spaces for various base schemes. The motivation for these two specific constructions is that Construction 3 is very intuitive, and a single-bit message space is frequently considered in the study of circular security.

Intuitively, the idea is to use an $n+1$-circular insecure encryption scheme Π to create an encryption scheme Π' so that a key of Π' consists of multiple keys in Π. This is done in a way so that from any length ℓ encryption cycle where $0 < \ell < n+1$ of Π', we can extract an encryption cycle with length $n+1$ of Π. We demonstrate this intuition by considering an encryption scheme Π with the following length 2

encryption cycle,

$$\mathcal{E}(\mathsf{pk}_0, \mathsf{sk}_1), \mathcal{E}(\mathsf{pk}_1, \mathsf{sk}_0).$$

We define a function f with two parameters in the following way:

$$f((\varphi_0, \varphi_1, \mathsf{s}),\ \psi) = (\mathcal{E}(\varphi_1, \psi), \mathsf{s}).$$

Let f be the encryption function for a scheme Π' where a public and a secret key takes the form $(\mathsf{pk}_0, \mathsf{pk}_1, \mathcal{E}(\mathsf{pk}_0, \mathsf{sk}_1))$ and sk_0 respecitvley. We observe that we can find a length 2 encryption cycle of Π in a length 1 cycle of Π':

$$f((\mathsf{pk}_0, \mathsf{pk}_1, \mathcal{E}(\mathsf{pk}_0, \mathsf{sk}_1)), \mathsf{sk}_0) = (\mathcal{E}(\mathsf{pk}_1, \mathsf{sk}_0), \mathcal{E}(\mathsf{pk}_0, \mathsf{sk}_1)).$$

7.1 Construction of n-Circular Insecure Encryption

We show how to construct a CPA secure ($\leq n$)-circular insecure public key encryption from a CPA secure $(n+1)$-circular insecure public key encryption. We present Constriction 3 in this section, which does not preserve message space. Our Construction 4, which preserves the message space of the bit encryption scheme will be presented in Section 7.3.

Construction 3. *Given a CPA secure $(n+1)$-circular insecure public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$. We obtain $\Pi' = (\mathcal{G}', \mathcal{E}', \mathcal{D}')$ as follows:*

Π' :

- $\mathcal{G}'(1^\eta)$: *Compute*

$$(\mathsf{pk}[0], \mathsf{sk}[0]) \leftarrow \mathcal{G}(1^\eta), \ldots, (\mathsf{pk}[n], \mathsf{sk}[n]) \leftarrow \mathcal{G}(1^\eta).$$

Compute $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[n])$. *For* $3 \leq i \leq n$ *compute* $h[i] \leftarrow \mathcal{E}(\mathsf{pk}[i], \mathsf{sk}[i-1])$. *Return public key*

$$\mathsf{pk} = \mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], h[n-1], \ldots, h[3].$$

and secret key

$$\mathsf{sk} = \mathsf{sk}[0], \ldots, \mathsf{sk}[n-1].$$

We note that when $n < 3$, the public key has nothing after s.

- $\mathcal{E}'(\mathsf{pk}, m)$: *Parse* pk *to*

$$\mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], \ldots, h[3].$$

If $|m| \neq |\mathsf{sk}|$, compute $c \leftarrow \mathcal{E}(\mathsf{pk}[1], m)$ and return

$$\langle 0, c \rangle.$$

Otherwise, parse m to $m[0], \ldots, m[n-1]$ where $|m[i]| = |\mathsf{sk}[i]|$ for $0 \leq i \leq n-1$. For $0 \leq i \leq n-1$ compute $c[i] \leftarrow \mathcal{E}(\mathsf{pk}[i+1], m[i])$, return

$$\langle 1, c[0], \ldots, c[n-1], \mathsf{s} \rangle.$$

- $\mathcal{D}'(\mathsf{sk}, c)$: *Parse* sk *to*

$$\mathsf{sk}[0], \ldots, \mathsf{sk}[n-1].$$

If $c = \langle 0, c' \rangle$, return $\mathcal{D}(\mathsf{sk}[1], c)$. Else parse c into $\langle 1, c[0], \ldots, c[n-1], \mathsf{s} \rangle$, compute $\mathsf{sk}[n] \leftarrow \mathcal{D}(\mathsf{sk}[0], \mathsf{s})$ and return

$$\mathcal{D}(\mathsf{sk}[1], c[0]) \parallel \ \ldots \ \parallel \mathcal{D}(\mathsf{sk}[n], c[n-1]).$$

One can observe that if all functions of Π are efficiently computable, so are the functions of Π'. Similarly, one can observe that if Π is correct, then so is Π'.

Remark 27. *We can delete the terms $h[\ldots]$ to achieve an $(n+1)$-circular insecurity to n-circular insecurity construction. Such a construction has reduced key size, and although the theoretical implications of our result are unchanged, it raised concerns about the key size blowup of going from $(n+1)$-circular insecurity to 1-circular insecurity.*

We provide the Figures 7.1 and 7.2 to visually represent how a length 3 encryption cycle of Π can be found in a length 1 and length 2 encryption cycle of Π' when $n = 2$. These examples are given in detail in Appendix A.4. These examples aim

to demonstrate the intuition behind Construction 3. Although the terms $h[\ldots]$ are not used in this example, they are used in the example of Section 7.2.1, where we set $n = 3$. The goal of the Section 7.2.1 is to illustrate the method that we use in the proof of Lemma 28.

Figure 7.1 visually represents how to spot a length 3 cycle of Π in a length 1 cycle of Π', which is $\mathcal{E}'(\mathsf{pk}, \mathsf{sk})$ for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}'(1^\eta)$. We interpret this graph by first observing that

$$\mathcal{E}'(\mathsf{pk}, \mathsf{sk}) = \langle 1, \mathcal{E}(\mathsf{pk}[1], \mathsf{sk}[0]), \mathcal{E}(\mathsf{pk}[2], \mathsf{sk}[1]), \mathsf{s} \rangle \text{ and } \mathsf{s} = \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[2]).$$

where $\mathsf{pk} = \mathsf{pk}[0], \mathsf{pk}[1], \mathsf{pk}[2], \mathsf{s}$ and $\mathsf{sk} = \mathsf{sk}[0], \mathsf{sk}[1]$. We will separate each element of this tuple with boxes. Then, following the directed arrows in the figure, each ciphertext is an encryption of the secret key corresponding to the public key used to make the next ciphertext.

Figure 7.1: 3-cycle of Π in 1-cycle of Π'

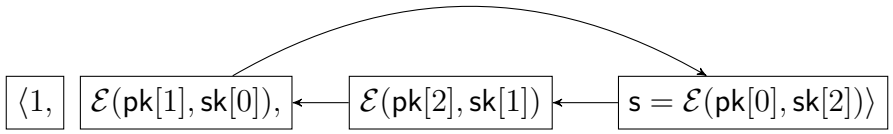


Figure 7.2 visually represents how to spot a length 3 cycle of Π in a length 2 cycle of Π', which is $\mathcal{E}'(\mathsf{pk}_0, \mathsf{sk}_1), \mathcal{E}'(\mathsf{pk}_1, \mathsf{sk}_0)$ given $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}'(1^\eta)$ and $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathcal{G}'(1^\eta)$. In the figure, we have

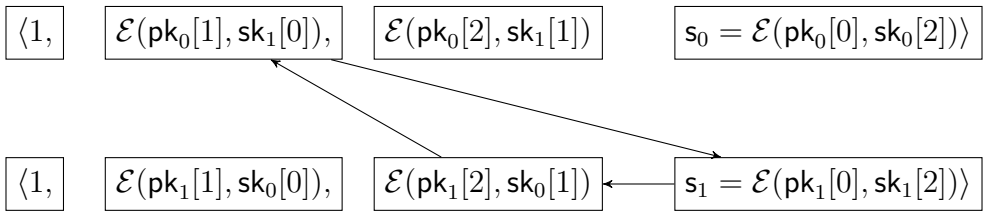
$$\mathcal{E}'(\mathsf{pk}_0, \mathsf{sk}_1) = \langle 1, \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_1[0]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_1[1]), \mathsf{s}_0 \rangle \text{ and } \mathsf{s}_0 = \mathcal{E}(\mathsf{pk}_0[0], \mathsf{sk}_0[2])$$

at the top row and

$$\mathcal{E}'(\mathsf{pk}_1, \mathsf{sk}_0) = \langle 1, \mathcal{E}(\mathsf{pk}_1[1], \mathsf{sk}_0[0]), \mathcal{E}(\mathsf{pk}_1[2], \mathsf{sk}_0[1]), \mathsf{s}_1 \rangle \text{ and } \mathsf{s}_1 = \mathcal{E}(\mathsf{pk}_1[0], \mathsf{sk}_1[2])$$

at the bottom row where for $i \in \{0, 1\}$, $\mathsf{pk}_i = \mathsf{pk}_i[0], \mathsf{pk}_i[1], \mathsf{pk}_i[2], \mathsf{s}_i$ and $\mathsf{sk}_i = \mathsf{sk}_i[0], \mathsf{sk}_i[1]$. Each tuple has its elements separated by boxes. Following the directed arrows, we have a length 3 encryption cycle of Π.

Figure 7.2: 3-cycle of Π in 2-cycle of Π'



It may be noted that the cycle following the directed edge is not the only way to extract a length 3 encryption cycle of Π from the ciphertexts above.

Theorem 5. *If a CPA secure $(n+1)$-circular insecure public key encryption scheme Π exists, then Π' is a CPA secure and $(\leq n)$-circular insecure encryption scheme.*

The following section shows this theorem via Lemma 28 and Lemma 32.

7.2 Analysis of Construction 3

This section will show that if Π is $(n+1)$-circular insecure and CPA secure, then Π' is $(\leq n)$-circular insecure and CPA secure.

We show Π' is $(\leq n)$-circular insecure by showing a way to find a length $n+1$ encryption cycle of Π in a length ℓ encryption cycle of Π', where $1 \leq \ell \leq n$. The proof of this is contained in Section 7.2.2, where we index the terms of the public keys and ciphertexts and show a way to find the length $n+1$ encryption cycle of Π. We provide an example in Section 7.2.1 that follows such an indexing method.

The proof of CPA security of Π' is presented in Section 7.2.3. We will use a sequence of games to show that the advantage of an adversary winning the CPA game against Π' is identical to that of an adversary winning the CPA game against Π except with negligible probability.

7.2.1 Circular Insecurity Example

In the following example, we consider the case where $n = 3$. We give an example of how to extract a length 4 encryption cycle of Π in a length 2 encryption cycle of Π', following the method presented in the proof of Lemma 28. Let $\mathsf{pk}_0, \mathsf{sk}_0 \leftarrow \mathcal{G}'(1^\eta)$ and

$\mathsf{pk}_1, \mathsf{sk}_1 \leftarrow \mathcal{G}'(1^\eta)$. For $i \in \{0,1\}$, we have

$$\mathsf{pk}_i = \mathsf{pk}_i[0], \mathsf{pk}_i[1], \mathsf{pk}_i[2], \mathsf{pk}_i[3], \mathsf{s}_i, h_i[3]$$

and

$$\mathsf{sk}_i = \mathsf{sk}_i[0], \mathsf{sk}_i[1], \mathsf{sk}_i[2].$$

Given the ciphertexts $c_0 = \mathcal{E}'(\mathsf{pk}_0, \mathsf{sk}_1)$ and $c[1] = \mathcal{E}'(\mathsf{pk}_1, \mathsf{sk}_0)$ and the public keys pk_0 and pk_1. For $i \in \{0,1\}$, let

$$c_i = \langle 1, c_i[0], c_i[1], c_i[2], \mathsf{s}_i \rangle.$$

Let $\ell = 2$ and $n = 3$. We can extract the following keys in the order described:

1. $\mathsf{pk}_{\ell-1}[0]$, which is $\mathsf{pk}_1[0]$.
2. For $\ell + 1 \leq i \leq n$ in desceinding order, $\mathsf{pk}_{\ell-1}[i]$. Which is $\mathsf{pk}_1[3]$.
3. For $0 \leq j \leq \ell - 1$ in ascending order, $\mathsf{pk}_{(\ell+j-1) \bmod \ell}[\ell - j]$. Which are $\mathsf{pk}_1[2]$, and $\mathsf{pk}_0[1]$.

So the keys we have extracted are $\mathsf{pk}_1[0], \mathsf{pk}_1[3], \mathsf{pk}_1[2], \mathsf{pk}_0[1]$, which are 4 keys of Π. We can also extract the following ciphertexts:

1. $\mathsf{s}_{\ell-1}$, which is s_1
2. For $\ell + 1 \leq i \leq n$ in descending order, $h_{\ell-1}[i]$. Which is $h_1[3]$.
3. For $0 \leq j \leq \ell - 1$ in ascending order, $c_{(\ell+j-1) \bmod \ell}[\ell - j - 1]$. Which are $c_1[1]$ and $c_0[0]$.

We observe that the ciphertexts extracted are the following, which form a length 4 encryption cycle of Π with the 4 keys extracted.

$$\begin{aligned}
\mathsf{s}_1 &\leftarrow \mathcal{E}(\mathsf{pk}_1[0], \mathsf{sk}_1[3]) \\
h_1[3] &\leftarrow \mathcal{E}(\mathsf{pk}_1[3], \mathsf{pk}_1[2]) \\
c_1[1] &\leftarrow \mathcal{E}(\mathsf{pk}_1[2], \mathsf{sk}_0[1]) \\
c_0[0] &\leftarrow \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_1[0]).
\end{aligned}$$

7.2.2 Circular Insecurity

Lemma 28. *If* Π *is* $(n+1)$*-circular insecure and CPA secure, then* Π' *is* $(\leq n)$*-circular insecure.*

Proof. Let Π be $(n+1)$-circular insecure and CPA secure. Then, there exists an adversary $\mathcal{A}$ such that

$$\Pr[\mathsf{CIRC}_{\mathcal{A},\Pi}(n+1,\eta)=1] \geq \frac{1}{2}+\epsilon(\eta).$$

Consider an arbitrary $\ell \in \{1,\ldots,n\}$. We can define an adversary $\mathcal{A}'_\ell$ as follows.

$\mathcal{A}'_\ell$:

1. Receive η and $\mathsf{pk}_0,\ldots,\mathsf{pk}_{\ell-1}$ where for $0 \leq x \leq \ell-1$

   $$\mathsf{pk}_x = \mathsf{pk}_x[0],\ldots,\mathsf{pk}_x[n],\mathsf{s}_x,h_x[n],\ldots,h_x[3].$$

   Send the following keys to $\mathcal{A}$ in the order described:

   (a) $\mathsf{pk}_{\ell-1}[0]$
   (b) for $\ell+1 \leq i \leq n$ in descending order, $\mathsf{pk}_{\ell-1}[i]$
   (c) for $0 \leq j \leq \ell-1$ in ascending order, $\mathsf{pk}_{(\ell+j-1) \bmod \ell}[\ell-j]$.

2. Receive $c^b_0,\ldots,c^b_{\ell-1}$ where for $0 \leq x \leq \ell-1$

   $$c^b_x = \langle 1, c^b_x[0],\ldots,c^b_x[n-1],\mathsf{s}_x\rangle.$$

   Send the following ciphertexts to $\mathcal{A}$ in the order described:

   (a) $\mathsf{s}_{\ell-1}$
   (b) for $\ell+1 \leq i \leq n$ in descending order, $h_{\ell-1}[i]$
   (c) for $0 \leq j \leq \ell-1$ in ascending order, $c^b_{(\ell+j-1) \bmod \ell}[\ell-j-1]$.

3. When $\mathcal{A}$ outputs b', output b'.

When $b=1$, we observe that

$$\mathsf{s}_{\ell-1} \leftarrow \mathcal{E}(\mathsf{pk}_{\ell-1}[0],\mathsf{sk}_{\ell-1}[n])$$

For $\ell + 1 \leq i \leq n$, (descending)

$$h_{\ell-1}[i] \leftarrow \mathcal{E}(\mathsf{pk}_{\ell-1}[i], \mathsf{sk}_{\ell-1}[i-1])$$

For $0 \leq j \leq \ell - 1$, (ascending)

$$c^1_{(\ell+j-1) \bmod \ell}[\ell - j - 1] \leftarrow \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod \ell}[\ell - j], \mathsf{sk}_{\ell+j \bmod \ell}[\ell - j - 1])$$

which is a $n + 1$ encryption cycle of Π, which means $\mathcal{A}'_\ell$ simulates $\mathsf{CIRC}_{\mathcal{A},\Pi}(n + 1, \eta)$ when the internal coin of $\mathsf{CIRC}_{\mathcal{A},\Pi}(n + 1, \eta)$ is 1.

When $b = 0$, we observe that

$$\mathsf{s}_{\ell-1} \leftarrow \mathcal{E}(\mathsf{pk}_{\ell-1}[0], \mathsf{sk}_{\ell-1}[n])$$

For $\ell + 1 \leq i \leq n$, (descending)

$$h_{\ell-1}[i] \leftarrow \mathcal{E}(\mathsf{pk}_{\ell-1}[i], \mathsf{sk}_{\ell-}[i-1])$$

For $0 \leq j \leq \ell - 1$, (ascending)

$$c^0_{(\ell+j-1) \bmod \ell}[\ell - j - 1] \leftarrow \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod \ell}[\ell - j], 0^{|\mathsf{sk}_0[0]|})$$

and by Lemma 15, this simulates $\mathsf{CIRC}_{\mathcal{A},\Pi}(n+1, \eta)$ when the internal coin of $\mathsf{CIRC}_{\mathcal{A},\Pi}(n+1, \eta)$ is 0 except with negligible probability.

This implies that if $\epsilon(\eta)$ is not negligible, then Π' is not ℓ-circular secure. Therefore we conclude if Π is $(n+1)$-circular insecure and CPA secure, then Π' is $(\leq n)$-circular secure. □

7.2.3 CPA security

Let Π be a CPA secure public key encryption scheme. We show Π' obtained from Π via Construction 3 is a CPA secure public key encryption scheme via a sequence of games [50].

Without loss of generality, we fix an efficient adversary $\mathcal{A}'$ to play in $\mathsf{CPA}_{\mathcal{A}',\Pi'}(\eta)$.

Game 0. This is just the game $\mathsf{CPA}_{\mathcal{A}',\Pi'}(\eta)$ with each step stated explicitly.

1. $(\mathsf{pk}[0], \mathsf{sk}[0]) \leftarrow \mathcal{G}(1^\eta), \ldots, (\mathsf{pk}[n], \mathsf{sk}[n]) \leftarrow \mathcal{G}(1^\eta)$.

Compute $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[n])$, for $3 \leq i \leq n$ compute $h[i] \leftarrow \mathcal{E}(\mathsf{pk}[i], \mathsf{sk}[i-1])$.
Give η and $\mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], \ldots, h[3]$ to $\mathcal{A}'$.

2. $b \leftarrow \$ \{0, 1\}$
3. $(m_0, m_1) \leftarrow \mathcal{A}'$.
4. If $m_b \neq |\mathsf{sk}[0], \ldots, \mathsf{sk}[n-1]|$, then compute $c^b \leftarrow \langle 0, \mathcal{E}(\mathsf{pk}[1], m_b)\rangle$.
   Otherwise,
   (a) Parse m_b into $m_b[0], \ldots, m_b[n-1]$.
   (b) For $0 \leq i \leq n-1$, compute $c^b[i] \leftarrow \mathcal{E}(\mathsf{pk}[i+1], m_b[i])$.
   Return $c_b = \langle 1, c^b[0], \ldots, c^b[n-1], \mathsf{s}\rangle$ to $\mathcal{A}'$.
5. $\mathcal{A}'$ outputs b'. The experiment outputs 1 if $b = b'$, 0 otherwise.

Let S_0 denote the event that Game 0. results in 1.

Game 1. Here, we modify how the public key is generated in step 1 of Game 0. Specically, we change how s and $h[n], \ldots, h[3]$ are computed in Game 1. Instead of encrypting $\mathsf{sk}[n], \ldots, \mathsf{sk}[2]$, they will encrypt strings zeros.

1. $(\mathsf{pk}[0], \mathsf{sk}[0]) \leftarrow \mathcal{G}(1^\eta), \ldots, (\mathsf{pk}[n], \mathsf{sk}[n]) \leftarrow \mathcal{G}(1^\eta)$.
   Compute $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0], 0^{|\mathsf{sk}[n]|})$, for $3 \leq i \leq n$ compute $h[i] \leftarrow \mathcal{E}(\mathsf{pk}[i], 0^{|\mathsf{sk}[i-1]|})$.
   Give η and $\mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], \ldots, h[3]$ to $\mathcal{A}'$.

Let S_1 denote the event that the result of Game 1. is 1.

Claim 29. *For some negligible function negl,*

$$|\Pr[S_0] - \Pr[S_1]| \leq negl(\eta).$$

Proof. We show this by applying Lemma 15. We define an adversary $\mathcal{A}$ that plays $\mathsf{ZERO}_{\mathcal{A},\Pi}(n+1, 2, \eta)$ as follows.

$\mathcal{A}$:

1. Receive η, $\mathsf{pk}_0, \ldots, \mathsf{pk}_n$ and $c_0^b, \ldots, c_n^b$.
2. Send the following to $\mathcal{A}'$ in the order described. $\mathcal{A}$ then sends the following in the order described to $\mathcal{A}'$:

(a) For $1 \leq i \leq n+1$ in descending order, $\mathsf{pk}_{i+1 \bmod (n+1)}$

(b) For $1 \leq i \leq n-1$ in ascending order, c_i^b.

What is being send above, depending on coin b, creates the public key, "$\mathsf{pk}[0], \ldots, \mathsf{pk}[n], s, h[n], \ldots, h[3]$", of either Game 0. or of Game 1.

3. Flip a coin d.
4. When $\mathcal{A}'$ queries two equal length challenge messages, m_0, m_1, if $|m_d| \neq n|\mathsf{sk}_0|$, compute $c^d \leftarrow \mathcal{E}(\mathsf{pk}_1, m_d)$ and send $\langle 0, c^d \rangle$ to $\mathcal{A}'$.
   Otherwise parse m_d to $m_d[0], \ldots, m_d[n-1]$, and for $0 \leq i \leq n-1$ in ascending order compute

$$c^d[i] \leftarrow \mathcal{E}(\mathsf{pk}_{n+1-i}, m^d[i]).$$

   Send $\langle 1, c^d[i], \ldots, c^d[n-1], c_1^b \rangle$ to $\mathcal{A}'$.
5. When $\mathcal{A}'$ outputs d', $\mathcal{A}$ outputs 1 if $d = d'$, output 0 otherwise.

Since each pk_i is generated by $\mathcal{G}$ of Π. We observe that in step 2 of the experiment, $\mathcal{A}'$ receives the following from $\mathcal{A}$:

- If $b = 1$:

$$\mathsf{pk}_1, \mathsf{pk}_0, \mathsf{pk}_n, \ldots, \mathsf{pk}_3, \mathsf{pk}_2, \mathcal{E}(\mathsf{pk}_1, \mathsf{sk}_2), \mathcal{E}(\mathsf{pk}_2, \mathsf{sk}_3), \ldots, \mathcal{E}(\mathsf{pk}_{n-1}, \mathsf{sk}_n).$$

  Which is identical to what is sent to $\mathcal{A}'$ in step 1 of Game 0.

- If $b = 0$:

$$\mathsf{pk}_1, \mathsf{pk}_0, \mathsf{pk}_n, \ldots, \mathsf{pk}_3, \mathsf{pk}_2, \mathcal{E}(\mathsf{pk}_1, 0^{|\mathsf{sk}|}), \mathcal{E}(\mathsf{pk}_2, 0^{|\mathsf{sk}|}), \ldots, \mathcal{E}(\mathsf{pk}_{n-1}, 0^{|\mathsf{sk}|}).$$

  Which is identical to what is sent to $\mathcal{A}'$ in step 1 of Game 1.

Therefore $\mathcal{A}$ simulates Game 0. when $b = 1$ and $\mathcal{A}$ simulates Game 1. when $b = 0$. This means when $b = 1$ $\mathcal{A}$ wins if $\mathcal{A}'$ loses, and when $b = 0$ $\mathcal{A}$ wins if $\mathcal{A}'$ wins. These

are the events $\neg S_0$ and S_1 respectively. Therefore, we have the following.

$$\begin{aligned}\Pr[\mathsf{ZERO}_{\mathcal{A},\Pi}(n,2,\eta)=1] &= \frac{1}{2}\Pr[\neg S_0] + \frac{1}{2}\Pr[S_1] \\ &= \frac{1}{2}(1+\Pr[S_1]-\Pr[S_0]) \\ &= \frac{1}{2}+\frac{1}{2}\left(\Pr[S_1]-\Pr[S_0]\right).\end{aligned}$$

Therefore if $|\Pr[S_0]-\Pr[S_1]|$ is not negligible, then by Lemma 15, Π is not CPA secure. □

Now, we define a series of games. The exact number of games depends on the cycle length n in the construction. This n does not depend on the security parameter η. We index a game in this series as Game 2.j for $j \geq 0$.

Game 2.j We let Game 2.0 be Game 1. In Game 2.j for $j > 0$, we modify how the query is answered in step 4 of Game 2.$(j-1)$.

4. If $m_b \neq |\mathsf{sk}[0],\ldots,\mathsf{sk}[n-1]|$, then compute $c^b \leftarrow \langle 0, \mathcal{E}(\mathsf{pk}[1], m_b)\rangle$. Otherwise,
   (a) Parse m_b into $m_b[0],\ldots,m_b[n-1]$.
   (b) For $0 \leq i \leq n-j-1$, compute $c^b[i] \leftarrow \mathcal{E}(\mathsf{pk}[i+1], m_b[i])$.
   (c) For $n-j \leq i \leq n-1$, compute $c^b[i] \leftarrow \mathcal{E}(\mathsf{pk}[i+1], 0^{|\mathsf{sk}[0]|})$.
   Return $c_b = \langle 1, c^b[0],\ldots,c^b[n-1]\rangle$ to $\mathcal{A}'$.

For $0 \leq j$, we let $S_{2.j}$ denote the event that Game 2.j. results in 1.

Claim 30. *For some negligible funciton negl,*

$$|\Pr[S_{2.j}] - \Pr[S_{2.(j+1)}]| \leq \mathit{negl}(\eta).$$

Proof. We defined an adversary $\mathcal{A}$ to play $\mathsf{CPA}_{\mathcal{A},\Pi}(\eta)$ as follows.

$\mathcal{A}$:

1. Receive η and pk. Label this pk as pk_{n-j}.
2. For $i \in \{0,\ldots,n\} \setminus \{n-j\}$, compute $\mathsf{pk}_i, \mathsf{sk}_i \leftarrow \mathcal{G}(1^\eta)$. Compute $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}_0, 0^{|\mathsf{sk}_0|})$. For $3 \leq i \leq n$, compute $h[i] \leftarrow \mathcal{E}(\mathsf{pk}_i, 0^{|\mathsf{sk}_0|})$. Send

$$\mathsf{pk}_0,\ldots,\mathsf{pk}_n,\mathsf{s},h[n],\ldots,h[3]$$

to $\mathcal{A}'$.

3. Flip a coin d.
4. When $\mathcal{A}$ receive two equal length challenge messages m_0, m_1 from $\mathcal{A}'$, if $|m_d| \neq n \cdot |\mathsf{sk}_0|$, compute $c^d \leftarrow \mathcal{E}(\mathsf{pk}[1], c^d)$ and send $\langle 0, c^d \rangle$ to $\mathcal{A}'$. Else create and query challenge messages $m'_0 = m_d[n-(j+1)]$ and $m'_1 = 0^{|m_d[0]|}$ to receive $c^b[n-(j+1)] \leftarrow \mathcal{E}(\mathsf{pk}_{n-j}, m'_b)$. For $i \in \{0, n-1\} \setminus \{n-(j+1)\}$ compute
$$c^d[i] = \begin{cases} \mathcal{E}(\mathsf{pk}[i+1], m[i]) & \text{if } i \leq n-(j+1) \\ \mathcal{E}(\mathsf{pk}[i+1], 0^{|m[i]|}) & \text{otherwise.} \end{cases}$$
Send $\langle 1, c^d[0], c^d[1], \ldots, c^d[n-1], \mathsf{s} \rangle$ to $\mathcal{A}'$.
5. When $\mathcal{A}'$ outputs d', $\mathcal{A}$ outputs 1 if $d = d'$, output 0 otherwise.

When $b = 0$, $\mathcal{A}$ simulates Game 2.j, and when $b = 1$, $\mathcal{A}$ simulates Game 2.$(j+1)$ when the internal coin is d. Therefore when $b = 0$, $\mathcal{A}$ wins if $d \neq d'$, which is the probability that $\mathcal{A}'$ fails in Game 2.j. When $b = 1$, $\mathcal{A}$ wins if $\mathcal{A}'$ wins in Game 2.$(j+1)$. This gives us the following:

$$\begin{aligned} \mathsf{CPA}_{\mathcal{A},\Pi}(\eta) &= \frac{1}{2}\Pr[\neg S_{2.j}] + \frac{1}{2}\Pr[S_{2.(j+1)}] \\ &= \frac{1}{2}(1 + \Pr[S_{2.(j+1)}] - \Pr[S_{2.j}]). \end{aligned}$$

Therefore, if $|\Pr[S_{2.j}] - \Pr[S_{2.(j+1)}]|$ is not negligible, then Π is not CPA secure. □

Now we show that $\mathcal{A}'$ can only win Game 2.n with negligible probability. This is because the game 2.n is essentially just a CPA game with Π.

Claim 31. *For some negligible function negl,*

$$\Pr[S_n] \leq \frac{1}{2} + negl(\eta).$$

Proof. Notice that in Game 2.n, when the lengths of the challenge messages are not equal to $n|\mathsf{sk}_0|$, then encryptions of zero are returned regardless of the coin of the game. When $\mathcal{A}$ queries challenge messages on other lengths, it receives a ciphertext encrypted by public key pk_1. Further, in Game 2.n, the key pk_1 is never used as the message of any ciphertext, which means Game 2.n can be simulated by any adversary

with a CPA oracle in a CPA game against Π. Therefore, if Π is CPA secure, then

$$\Pr[S_n] \leq \frac{1}{2} + \mathrm{negl}(\eta).$$

□

Lemma 32. *If* Π *is CPA secure, then* Π' *is CPA secure.*

Proof. This is shown via the sequence of games above. Assume Π is CPA secure. Let $\mathrm{negl}_0(\eta)$ denote the difference beween $\Pr[S_0]$ and $\Pr[S_1]$. Let $\mathrm{negl}_{i+1}(\eta)$ denote the difference between $\Pr[S_{2.i}]$ and $\Pr[S_{2.(i+1)}]$. Let negl_{n+1} denote the advatange of $\mathcal{A}'$ in CPA against Π. Then we have

$$\begin{aligned}
\mathsf{CPA}_{\mathcal{A}',\Pi'}(\eta) = \Pr[S_0] &\leq \Pr[S_{2.0}] + \mathrm{negl}_0(\eta) \\
&\leq \Pr[S_{2.1}] + \sum_{i\in\{0,1\}} \mathrm{negl}_i(\eta) \\
&\vdots \\
&\leq \Pr[S_{2.n}] + \sum_{i\in\{0,\ldots,n\}} \mathrm{negl}_i(\eta) \\
&\leq \frac{1}{2} + \sum_{i\in\{0,\ldots,n+1\}} \mathrm{negl}_i(\eta).
\end{aligned}$$

This concludes that Π' is CPA secure. □

7.3 Bit Encryption Construction

In this section, we present a construction that preserves the message space of a base scheme. Construction 4 differs from Construction 3 in the encryption and decryption functions. In Construction 4, we are always expecting the message to have length 1, and we will create n ciphertexts using n of the $n+1$ public keys.

Construction 4. *Given a CPA secure $(n+1)$-circular insecure public key bit encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$. We obtain $\Pi^{bit} = (\mathcal{G}^{bit}, \mathcal{E}^{bit}, \mathcal{D}^{bit})$ as follows.*

Π^{bit}*:*

- $\mathcal{G}^{bit}$*: Compute*

$$(\mathsf{pk}[0], \mathsf{sk}[0]) \leftarrow \mathcal{G}(1^\eta), \ldots (\mathsf{pk}[n], \mathsf{sk}[n]) \leftarrow \mathcal{G}(1^\eta).$$

  Compute $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[n])$ *(one bit at a time). For* $3 \leq i \leq n$ *compute* $h[i] \leftarrow \mathcal{E}(\mathsf{pk}[i], \mathsf{sk}[i-1])$ *(one bit at a time). Return public key*

$$\mathsf{pk} = \mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], h[n-1], \ldots, h[3].$$

  and secret key

$$\mathsf{sk} = \mathsf{sk}[0], \ldots, \mathsf{sk}[n-1].$$

- $\mathcal{E}^{bit}(pk, m)$*: Note that* $m \in \{0,1\}$*. Parse* pk *to*

$$\mathsf{pk}[0], \ldots, \mathsf{pk}[n], \mathsf{s}, h[n], \ldots, h[3].$$

  For $0 \leq i \leq n-1$*. For* $0 \leq i \leq n-1$ *compute* $c[i] \leftarrow \mathcal{E}(\mathsf{pk}[i+1], m)$*, return*

$$c[0], \ldots, c[n-1]$$

- $\mathcal{D}^{bit}(\mathsf{sk}, c)$: *Parse* c *into* $c[0], \ldots, c[n-1]$*, return*

$$\mathcal{D}(\mathsf{sk}[1], c[0]).$$

Construction 4's proof of CPA security is similar to that of Construction 3. We can repeat the sequence of games from Section 7.2.3 with some modifications. In each encryption, instead of parsing m to n strings each of length $|\mathsf{sk}[0]|$, we will encrypt

the message bit m with n 'sub'-public keys. In modified Game $2.n$, any key used to encrypt a message does not appear as a message anywhere, and this game is similar to a multiple-key CPA game. A multiple-key CPA game, as its name suggests, is a game played with multiple keys, each encrypting the same challenge message. [1] It can be easily observed that an encryption scheme is CPA secure if and only if such a scheme is multiple-key CPA secure.

Before we proceed to show that the encryption scheme is $(\leq n)$-circular insecure, we provide an example.

Example 33. *In this simplified example, we assume $n = 2$, and the secret key of the base scheme Π has length 2. A length 1 encryption cycle of Π^{bit} with $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}^{bit}(\eta)$ is the following*

$$
\begin{aligned}
&\mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0) \\
&= \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[0][0]), \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[0][1]), \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[1][0]), \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[1][1]) \\
&= \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[0][0]), \underline{\mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][0])}, \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[0][1]), \underline{\mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][1])}, \\
&\qquad \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[1][0]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[1][0]), \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[1][1]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[1][1]),
\end{aligned}
$$

where $\mathsf{pk}_0 = \mathsf{pk}_0[0], \mathsf{pk}_0[1], \mathsf{pk}_0[2], \mathsf{s}_0$ and $\mathsf{sk}_0 = \mathsf{sk}_0[0], \mathsf{sk}_0[1]$. For $i, j \in \{0, 1\}$, $\mathsf{sk}_0[i][j]$ denotes the jth bit of $\mathsf{sk}_0[i]$. Note that in this example, we assume $|\mathsf{sk}_0[i]| = 2$. The reason for the expansions above is because Π^{bit} and Π are both bit encryptions, which encrypt messages with lengths longer than 1 bit by bit. If we extract just the underlined terms, we have

$$\mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][0]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][1]) = \mathcal{E}(\mathsf{pk}_0[2])(\mathsf{sk}_0[0]).$$

It can be observed that $\mathsf{s}_0 \leftarrow \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[2])$. Combining the underlined terms with s we have $\mathcal{E}(\mathsf{pk}[2], \mathsf{sk}[0]), \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[2])$ which is a length 2 circular expression of Π.

Now, to show this scheme is $(\leq n)$-circular insecure. We remind the readers that for bit encryption, when we say $\mathcal{E}(k, m)$ for $|m| > 1$, we mean encrypting each bit of m in the order they appear in m, each time with fresh randomness. Consider an arbitrary $\ell \in \{1, \ldots, n\}$. Given $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}^{\mathsf{bit}}(1^\eta), \ldots, (\mathsf{pk}_\ell, \mathsf{sk}_\ell) \leftarrow \mathcal{G}^{\mathsf{bit}}(1^\eta)$, for

[1] We provide a definition of multiple-key CPA in Appendix A.3.

$0 \leq i < \ell$ let

$$c_i^b \leftarrow \begin{cases} \mathcal{E}^{\mathsf{bit}}(\mathsf{pk}_i, \mathsf{sk}_{i+1 \bmod \ell}) & \text{if } b = 1 \\ \mathcal{E}^{\mathsf{bit}}(\mathsf{pk}_i, 0^{|\mathsf{sk}_0|}) & \text{otherwise.} \end{cases}$$

To keep the notation tidy, let $p = |\mathsf{sk}_0[0]|$.

An adversary $\mathcal{A}_\ell^{\mathsf{bit}}$ in the game $\mathsf{CIRC}_{\mathcal{A}^{\mathsf{bit}},\Pi^{\mathsf{bit}}}(\ell, \eta)$ after receiving $c_0^b, \ldots, c_{\ell-1}^b$ can do the following. First for $0 \leq i < \ell$, parse c_i^b into

$$c_i^b[0], \ldots, c_i^b[n-1],$$

where $c_i^b[j]$ for $0 \leq j \leq n-1$ is sampled from $\mathcal{E}^{\mathsf{bit}}(\mathsf{pk}_i, m_b)$ where $m_1 = \mathsf{sk}_{i+1 \bmod \ell}[j]$ and $m_0 = 0^p$.
For $0 \leq j \leq n-1$, $c_i^b[j]$ can be parsed into

$$c_i^b[j][0], c_i^b[j][1], \ldots, c_i^b[j][p-1]$$

where for $0 \leq x \leq p$, $c_i^b[j][x]$ is samples from $\mathcal{E}^{\mathsf{bit}}(\mathsf{pk}_i, m'_b)$ where $m'_1 = \mathsf{sk}_{i+1 \bmod \ell}[j][x]$ (the xth bit of $\mathsf{sk}_{i+1 \bmod \ell}[j]$) and $m'_0 = 0$.

For $0 \leq x \leq p-1$, $c_i^b[j][x]$ can be parsed to

$$c_i^b[j][x][0], \ldots, c_i^b[j][x][n-1]$$

where for $0 \leq y \leq n-1$, $c_i^b[i][x][y]$ is sampled from $\mathcal{E}(\mathsf{pk}_i[y+1], m''_b)$ where $m''_1 = \mathsf{sk}_{i+1 \bmod \ell}[j][x]$ and $m''_0 = 0$.

Remark 34. *Putting what we have just described into Example 33, we have*

$$\begin{aligned}
c_0^1 &\leftarrow \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0) \\
c_0^1[0] &\leftarrow \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[0]) \\
c_0^1[1] &\leftarrow \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[1]) \\
c_0^1[0][0] &\leftarrow \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[0][0]) = \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[0][0]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][0]) \\
c_0^1[0][1] &\leftarrow \mathcal{E}^{bit}(\mathsf{pk}_0, \mathsf{sk}_0[0][1]) = \mathcal{E}(\mathsf{pk}_0[1], \mathsf{sk}_0[0][1]), \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][1]) \\
\underline{c_0^1[0][0][1]} &\leftarrow \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][0]) \\
\underline{c_0^1[0][1][1]} &\leftarrow \mathcal{E}(\mathsf{pk}_0[2], \mathsf{sk}_0[0][1])
\end{aligned}$$

The underlined terms correspond to the underlined terms in Example 33.

Now that the ciphertext has been parsed, the adversary can extract the following in the order described:

1. $\mathsf{s}_{\ell-1}$
2. for $\ell+1 \leq i \leq n$ in descending order $h_{\ell-1}[i]$
3. for $0 \leq j \leq \ell-1$ in ascending order:
   (a) for $0 \leq x \leq p-1$ in ascending order: $c^b_{(\ell+j-1) \bmod \ell}[\ell-j-1][x][\ell-j-1]$.

Since

$$
\begin{aligned}
&c^b_{(\ell+j-1) \bmod \ell}[\ell-j-1][0][\ell-j-1], \ldots, c^b_{(\ell+j-1) \bmod \ell}[\ell-j-1][p-1][\ell-j-1] \\
&\leftarrow \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod i}[\ell-j], \mathsf{sk}_{(\ell+j) \bmod \ell}[\ell-j-1][0]), \\
&\quad \vdots \\
&\quad \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod i}[\ell-j], \mathsf{sk}_{(\ell+j) \bmod \ell}[\ell-j-1][p-1]) \\
&= \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod i}[\ell-j], \mathsf{sk}_{(\ell+j) \bmod \ell}[\ell-j-1]).
\end{aligned}
$$

The ciphertexts that the adversary has extracted when $b = 1$ are sampled from:

1. $s_{\ell_1} \leftarrow \mathcal{E}(\mathsf{pk}_\ell[0], \mathsf{sk}_\ell[n])$
2. for $\ell+1 \leq i < n$ in descending order, $h_{\ell-1}[i] \leftarrow \mathcal{E}(\mathsf{pk}_{\ell-1}[i], \mathsf{sk}_{\ell-1}[i-1])$
3. for $0 \leq j \leq \ell-1$ in ascending order,

$$c^1_{(\ell+j-1) \bmod \ell}[\ell-1-j][0,\ldots,p-1][\ell-j-1] \leftarrow \mathcal{E}(\mathsf{pk}_{(\ell+j-1) \bmod i}[\ell-j], \mathsf{sk}_{(\ell+1) \bmod \ell}[\ell-j-1]).$$

which is a $n+1$-cycle with the following keys in Π,

1. $\mathsf{pk}_{\ell-1}[0], \mathsf{sk}_{\ell-1}[0]$
2. for $\ell+ \leq i < n$, $\mathsf{pk}_{\ell-1}[i], \mathsf{sk}_{\ell-1}[i]$
3. for $0 \leq j \leq \ell-1$, $\mathsf{pk}_{(\ell+j-1) \bmod \ell}[\ell-j], \mathsf{sk}_{(\ell+j-1) \bmod \ell}[\ell-j]$.

And in the event where $b = 0$, the ciphertexts extracted by $\mathcal{A}^{\mathsf{bit}}$ are indistinguishable from encryptions of zeros since it is only "almost a cycle" as in Lemma 15.

7.3.1 Arbitrary Message Space

To modify an $(n+1)$-circular insecure encryption scheme Π with message length β using Construction 4 and preserve the same message space, we first define a scheme $\bar{\Pi}$ where each key is padded with extra bits at the end so that the length of a key in $\bar{\Pi}$ is divisible by β. This step depends on how the encryption function of the base scheme pads the plaintext when it is shorter than its message space. Now, with the appropriate changes for calls to the encryption function of the base scheme, by encrypting β bits at a time using the same n keys as the bit version, Construction 4 is ready to be used.

7.4 Symmetric Encryption Scheme

This section discusses how to modify the constructions to work in the symmetric key setting. There are three changes to be made. The first change is to handle the fact that the public key is the secret key. The key generation function outputs all $n+1$ base secret keys. The terms s and $h[\ldots]$ are no longer computed by the key generation function; now they will be computed by the encryption function. The second change requires the encryption function to perform the computations of s and $h[n], \ldots h[3]$ and append the results as part of the ciphertext. i.e., each time the encryption function computes $\mathsf{s} \leftarrow \mathcal{E}(\mathsf{sk}[0], \mathsf{sk}[n]), h[n] \leftarrow \mathcal{E}(\mathsf{sk}[n], \mathsf{sk}[n-1]), \ldots, h[3] \leftarrow \mathcal{E}(\mathsf{sk}[3], \mathsf{sk}[2])$ and appends these ciphertexts at the end of the ciphertext of the message. The third is, if using Construction 3, during encryption, when expecting a message of the same length as the key, the function needs to perform an extra encryption to accommodate the new n-th key (zero-indexing). This encryption can be performed using sub-key $\mathsf{sk}[1]$.

We show what Construction 3 would look like in the symmetric key setting. Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ by a CPA secure and $(n+1)$-circular insecure. We can define $\Pi^{\mathsf{sym}} = (\mathcal{G}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}})$.

- $\mathcal{G}^{\mathsf{sym}}(1^\eta)$: $\mathsf{sk}[0] \leftarrow \mathcal{G}(1^\eta), \ldots, \mathsf{sk}[n] \leftarrow \mathcal{G}(1^\eta)$. Return $\mathsf{sk}^{\mathsf{sym}} = \mathsf{sk}[0], \ldots, \mathsf{sk}[n]$.
- $\mathcal{E}^{\mathsf{sym}}(\mathsf{sk}^{\mathsf{sym}}, m)$: if $|m| \neq |\mathsf{sk}^{\mathsf{sym}}|$, then return $\langle 0, \mathcal{E}(\mathsf{sk}[1], m)\rangle$. Else, parse m to $m[0], \ldots, m[n]$. Let $\mathsf{s}, h[n], \ldots, h[3]$ be $\mathcal{E}(\mathsf{sk}[0], \mathsf{sk}[n]), \mathcal{E}(\mathsf{sk}[n], \mathsf{sk}[n-1]), \ldots, \mathcal{E}(\mathsf{sk}[3], \mathsf{sk}[2])$. Return $\langle 1, \mathcal{E}(\mathsf{sk}[1], m[0]), \ldots, \mathcal{E}(\mathsf{sk}[n], m[n-1]), \mathcal{E}(\mathsf{sk}[1], m[n]), \mathsf{s}, h[n], \ldots, h[3]\rangle$.

- $\mathcal{D}^{\mathsf{sym}}(\mathsf{sk}^{\mathsf{sym}}, \langle d, c' \rangle)$: if $d = 0$, return $\mathcal{D}(\mathsf{sk}[1], m)$. Else pase c' to $c[0], \ldots, c[n]$, s, $h[n], \ldots, h[3]$. Return

$$\mathcal{D}(\mathsf{sk}[1], c[0]) \| \mathcal{D}(\mathsf{sk}[2], c[1]) \| \ldots \| \mathcal{D}(\mathsf{sk}[n], c[n-1]) \| \mathcal{D}(\mathsf{sk}[1], c[n]]).$$

The proof of security is not too different from the public key setting. The only difference is in Game 2.1, we are replacing both $c[0]$ and $c[n]$ with encryptions of zeros. To show the probability that Game 2.1 results in 1 is close to the probability that Game 2.0 or Game 2.2 result in 1, we can reduce the difference between the probabilities into the advantage of an adversary winning left-right oracle game [2] against the base encryption scheme Π. It is well known that an encryption scheme is LR-oracle secure if and only if it is CPA secure.

[2] See LR-oracle experiment in [31].

Chapter 8

Conclusion and Future Work

In this thesis, we have shown that Abadi-Rogaway logic is complete with respect to KDM security in Chapter 4, and Micciancio's logic is complete for more than just acyclic expressions in Chapter 5. Further, we characterized the equivalence relation in Micciancio's logic in Chapter 6 and discovered new implications for circular insecurity in Chapter 7.

Combining our results with prior results, we know Abadi-Rogaway logic is sound and complete with respect to KDM. This provides security guarantees when designing programs/protocols symbolically with KDM primitives. Unfortunately, it is uncertain if such convenience exists for Micccianciо's logic since its completeness with respect to CPA is not proven. However, our results bring us a step closer to this goal.

Although our results make progress towards finding a completeness for Micciancio's logic, the problem remains open. It would provide interesting theoretical implications if completeness is proved or disproved in the future.

Bibliography

[1] Martín Abadi and Andrew D Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 36–47, 1997.

[2] Martí;n Abadi and Phillip Rogaway. Reconciling two views of cryptography. *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, page 3–22, 2000.

[3] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 403–422. Springer, 2010.

[4] Pedro Adao, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness and completeness of formal encryption: The cases of key cycles and partial information leakage. *Journal of Computer Security*, 17(5):737–797, 2009.

[5] Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (ring-) lwe. In *Annual International Cryptology Conference*, pages 659–680. Springer, 2016.

[6] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. *Journal of cryptology*, 27(3):429–451, 2014.

[7] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 423–444. Springer, 2010.

[8] Allison Bishop, Susan Hohenberger, and Brent Waters. New circular security counterexamples from decision linear and learning with errors. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 776–800. Springer, 2015.

[9] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002 St. John's, Newfoundland, Canada, August 15–16, 2002 Revised Papers 9*, pages 62–75. Springer, 2003.

[10] Bruno Blanchet. Security protocol verification: Symbolic and computational models. In *International conference on principles of security and trust*, pages 3–29. Springer, 2012.

[11] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 227–240. 2019.

[12] Dan Boneh, Shai Halevi, Mike Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology–CRYPTO 2008: 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings 28*, pages 108–125. Springer, 2008.

[13] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, pages 201–218. Springer, 2011.

[14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.

[15] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems (TOCS)*, 8(1):18–36, 1990.

[16] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In *International Workshop on Public Key Cryptography*, pages 540–557. Springer, 2012.

[17] Véronique Cortier and Eugen Zălinescu. Deciding key cycles for security protocols. In *Logic for Programming, Artificial Intelligence, and Reasoning: 13th International Conference, LPAR 2006, Phnom Penh, Cambodia, November 13-17, 2006. Proceedings 13*, pages 317–331. Springer, 2006.

[18] Francien Dechesne and Yanjing Wang. To know or not to know: epistemic approaches to security protocol verification. *Synthese*, 177:51–76, 2010.

[19] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[20] F Javier Thayer Fábrega, Jonathan C Herzog, and Joshua D Guttman. Strand spaces: Why is a security protocol correct? In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186)*, pages 160–171. IEEE, 1998.

[21] Marcelo Fiore and Martin Abadi. Computing symbolic models for verifying cryptographic protocols. In *Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001.*, pages 160–173. IEEE, 2001.

[22] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2021.

[23] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, 2013.

[24] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[25] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 528–557. Springer, 2017.

[26] James W Gray and John McLean. Using temporal logic to specify and verify cryptographic protocols. In *Proceedings The Eighth IEEE Computer Security Foundations Workshop*, pages 108–116. IEEE, 1995.

[27] Matthew Green and Susan Hohenberger. Cpa and cca-secure encryption systems that are not 2-circular secure, 2010.

[28] Mohammad Hajiabadi and Bruce M Kapron. Computational soundness of coinductive symbolic security under active attacks. In *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 539–558. Springer, 2013.

[29] Mohammad Hajiabadi and Bruce M Kapron. Toward fine-grained blackbox separations between semantic and circular-security notions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 561–591. Springer, 2017.

[30] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235. IEEE Computer Society, 1989.

[31] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[32] Richard Kemmerer, Catherine Meadows, and Jonathan Millen. Three systems for cryptographic protocol analysis. *Journal of CRYPTOLOGY*, 7:79–130, 1994.

[33] Richard A Kemmerer. Analyzing encryption protocols using formal verification techniques. *IEEE Journal on Selected areas in Communications*, 7(4):448–457, 1989.

[34] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for kdm security. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, pages 253–285. Springer, 2020.

[35] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In *Theory of Cryptography: 12th Theory*

of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12, pages 378–400. Springer, 2015.

[36] Peeter Laud. Encryption cycles and two views of cryptography. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC)*, volume 31, pages 85–100. Citeseer, 2002.

[37] Baiyu Li and Daniele Micciancio. Symbolic security of garbled circuits. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 147–161. IEEE, 2018.

[38] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, 1996.

[39] Antonio Marcedone and Claudio Orlandi. Obfuscation $\rightarrow$ (ind-cpa security $\not\rightarrow$ circular security). In *International Conference on Security and Cryptography for Networks*, pages 77–90. Springer, 2014.

[40] Catherine Meadows. A system for the specification and verification of key management protocols. In *Proceedings. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 182–182. IEEE Computer Society, 1991.

[41] Daniele Micciancio. Computational soundness, co-induction, and encryption cycles. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 362–380. Springer, 2010.

[42] Daniele Micciancio. Symbolic encryption with pseudorandom keys. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*, pages 64–93. Springer, 2019.

[43] Daniele Micciancio and Saurabh Panjwani. Adaptive security of symbolic encryption. In *Theory of Cryptography Conference*, pages 169–187. Springer, 2005.

[44] Daniele Micciancio and Bogdan Warinschi. Completeness theorems for the abadi–rogaway language of encrypted expressions1. *Journal of Computer Security*, 12(1):99–129, 2004.

[45] Microsoft, 2024. https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/faq#key-management.

[46] Jonathan K. Millen, Sidney C. Clark, and Sheryl B. Freedman. The interrogator: Protocol secuity analysis. *IEEE Transactions on software Engineering*, (2):274–288, 1987.

[47] John C Mitchell, Mark Mitchell, and Ulrich Stern. Automated analysis of cryptographic protocols using mur/spl phi. In *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097)*, pages 141–151. IEEE, 1997.

[48] Lawrence C Paulson. The inductive approach to verifying cryptographic protocols. *Journal of computer security*, 6(1-2):85–128, 1998.

[49] Ron D Rothblum. On the circular security of bit-encryption. In *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 579–598. Springer, 2013.

[50] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs, 2004. *URL: http://eprint. iacr. org/2004/332*, 2006.

[51] Brent Waters and Daniel Wichs. Universal amplification of kdm security: From 1-key circular to multi-key kdm. In *Annual International Cryptology Conference*, pages 674–693. Springer, 2023.

[52] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.

Appendix A

Additional Information

A.1 Circular Counter-Example

Consider the following expressions to demonstrate that we need circular counter-examples for many different lengths.

$$E_0 = \{\{\{K_0\}_{K_0}\}_{K_1}, \{K_1\}_{K_2}, \{K_2\}_{K_3}, \{K_3\}_{K_2}$$
$$E_0 = \{\{\{K_4\}_{K_0}\}_{K_1}, \{K_1\}_{K_2}, \{K_2\}_{K_3}, \{K_3\}_{K_2}$$

where each $K_i, i \in \{0, \ldots, 4\}$ is unique. We can observe that $E_0 \not\sim_{\text{FIX}} E_1$.

Let Π be a CPA secure encryption scheme. If Π is only 1-circular insecure or only 2-circular insecure, it is not sufficient for the evaluations of these two expressions to be distinguishable.

However, if for some CPA-secure encryption scheme Π^3, there is an adversary that can recover a secret key from a length 3 cycle of Π^3, then we can define an encryption scheme $\Pi^{(3)}$ and adversary $\mathcal{A}^{(3)}$ such that it can distinguish $[\![E_0]\!]_{\Pi^{(3)}}$ from $[\![E_1]\!]_{\Pi^{(3)}}$.

A.2 Folklore Constructions for Length 1 Cycle

Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a CPA/CCA-secure encryption scheme. Let f be an independent one-way function.

We present folklore constructions in the public key setting, which can be modified minimally to work in the symmetric key setting. We assume Π's message space is equal to its key space. When the message space is larger, we can pad the secret key

with random bits to achieve the same space. Our presentation may differ from the presentations of others. Still, the core idea of the folklore construction is using a base scheme as a black box to construct a new scheme so that the new encryption function "checks" if the message is the secret key to the public key.

We can build an 1-circular secure encryption scheme $\Pi^{\mathtt{1\text{-}sec}} = (\mathcal{G}^{\mathtt{1\text{-}sec}}, \mathcal{E}^{\mathtt{1\text{-}sec}}, \mathcal{D}^{\mathtt{1\text{-}sec}})$ from Π as follows:

- $\mathcal{G}^{\mathtt{1\text{-}sec}}(1^\eta)$: $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}(1^\eta)$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathcal{G}(1^\eta)$.
  Let $\mathsf{pk}^{\mathtt{1\text{-}sec}} = (f(\mathsf{sk}_1), \mathsf{pk}_0)$ and $\mathsf{sk}^{\mathtt{1\text{-}sec}} = (\mathsf{sk}_0, \mathsf{sk}_1)$.
  Return $\mathsf{pk}^{\mathtt{1\text{-}sec}}$ and $\mathsf{sk}^{\mathtt{1\text{-}sec}}$.

- $\mathcal{E}^{\mathtt{1\text{-}sec}}(\mathsf{pk}^{\mathtt{1\text{-}sec}}, m)$: Parse $\mathsf{pk}^{\mathtt{1\text{-}sec}}$ to $(f(\mathsf{sk}_1), \mathsf{pk}_0)$.
  If $f(m) = f(\mathsf{sk}_1)$, return $(\mathcal{E}(\mathsf{pk}_0, 1), \mathcal{E}(\mathsf{pk}, r))$ where $r \leftarrow \$\,\{0,1\}^{|m|}$.
  Otherwise, return $(\mathcal{E}(0), \mathcal{E}(\mathsf{pk}_0, m))$.

- $\mathcal{D}^{\mathtt{1\text{-}sec}}(\mathsf{sk}^{\mathtt{1\text{-}sec}}, c)$: Parse c to (c_0, c_1) and $\mathsf{sk}^{\mathtt{1\text{-}sec}}$ to $(\mathsf{sk}_0, \mathsf{sk}_1)$
  If $\mathcal{D}(\mathsf{sk}_0, c_0) = 1$, return $(\mathsf{sk}_0, \mathsf{sk}_1)$.
  Otherwise, return $\mathcal{D}(\mathsf{sk}_0, c_1)$.

This encryption scheme has only overwhelming correctness instead of full correctness.

To create a 1-circular insecure encryption, we can create $\Pi^{\mathtt{1\text{-}ins}} = (\mathcal{G}^{\mathtt{1\text{-}ins}}, \mathcal{E}^{\mathtt{1\text{-}ins}}, \mathcal{D}^{\mathtt{1\text{-}ins}})$:

- $\mathcal{G}^{\mathtt{1\text{-}ins}}(1^\eta)$: $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}(1^\eta)$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathcal{G}(1^\eta)$.
  Let $\mathsf{pk}^{\mathtt{1\text{-}ins}} = (f(\mathsf{sk}_1), \mathsf{pk}_0)$ and $\mathsf{sk}^{\mathtt{1\text{-}ins}} = (\mathsf{sk}_0, \mathsf{sk}_1)$.
  Return $\mathsf{pk}^{\mathtt{1\text{-}ins}}$ and $\mathsf{sk}^{\mathtt{1\text{-}ins}}$.

- $\mathcal{E}^{\mathtt{1\text{-}ins}}(\mathsf{pk}^{\mathtt{1\text{-}ins}}, m)$: Parse $\mathsf{pk}^{\mathtt{1\text{-}ins}}$ to $(f(\mathsf{sk}_1), \mathsf{pk}_0)$.
  If $f(m) = f(\mathsf{sk}_1)$, return $(1, \mathcal{E}(\mathsf{pk}_0, m))$.
  Otherwise, return $(0, \mathcal{E}(\mathsf{pk}_0, m))$.

- $\mathcal{D}^{\mathtt{1\text{-}ins}}(\mathsf{sk}^{\mathtt{1\text{-}ins}}, c)$: Parse c to (b, c') and $\mathsf{sk}^{\mathtt{1\text{-}ins}}$ to $(\mathsf{sk}_0, \mathsf{sk}_1)$
  Return $\mathcal{D}(\mathsf{sk}_0, c')$.

A.3 Multiple-Key CPA security

In the following, we assume n is a non-zero polynomial of η.

Definition 35 (Multiple-Key CPA)**.** *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme. The n-key CPA experiment for Π and adversary $\mathcal{A}$ denoted by $\mathsf{multi\text{-}CPA}_{\mathcal{A},\Pi}(\eta, n)$ is defined as follows:*

$\mathit{multi\text{-}CPA}_{\mathcal{A},\Pi}(\eta, n)$:

1. $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathcal{G}(1^\eta), \ldots, (\mathsf{pk}_{n-1}, \mathsf{sk}_{n-1}) \leftarrow \mathcal{G}(1^\eta)$. *Give* η *and* $\mathsf{pk}_0, \ldots, \mathsf{pk}_n$ *to* $\mathcal{A}$.
2. $b \leftarrow \$ \{0, 1\}$.
3. $\mathcal{A}$ *outputs two equal length challenge messages* m_0, m_1.
4. *Compute* $c_i^b \leftarrow \mathcal{E}(\mathsf{pk}_i, m_b)$ *for* $i \in \{0, \ldots, n-1\}$ *and return* $c_0^b, \ldots, c_{n-1}^b$ *to* $\mathcal{A}$.
5. $\mathcal{A}$ *outputs* b' *and the experiment results in* 1 *if* $b = b'$, 0 *otherwise.*

We say Π *is n-key CPA secure if for every PPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathit{multi\text{-}CPA}_{\mathcal{A},\Pi}(\eta, n) = 1] \leq \frac{1}{2} + \mathit{negl}(\eta).$$

A.4 Details Figure 2 and 3

Consider $n = 2$. Let Π be a 3-circular insecure and CPA secure public key encryption scheme, and suppose Π' is constructed from Π via Construction 3. We can show that Π' is both 1-circular insecure and 2-circular insecure.

Fix a security parameter η. Consider some adversary $\mathcal{A}$ such that

$$\Pr[\mathsf{CIRC}_{\mathcal{A},\Pi}(3, \eta) = 1] = \frac{1}{2} + \epsilon(\eta).$$

To see Π' is 1-circular insecure, let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}'(1^\eta)$. Define adversary $\mathcal{A}_1'$ as follows:

$\mathcal{A}_1'$:

1. Receive η and $\mathsf{pk} = \mathsf{pk}[0], \mathsf{pk}[1], \mathsf{pk}[2], \mathsf{s}$. Send $\mathsf{pk}[0], \mathsf{pk}[2], \mathsf{pk}[1]$ to $\mathcal{A}$.
2. Receive $c^b = \langle 1, c^b[0], c^b[1] \rangle$. Send $\mathsf{s}, c^b[1], c^b[0]$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs b', output b'.

In the event $b = 1$, $\mathcal{A}_1'$ has sent ciphertexts

$$\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0], \mathsf{sk}[2]) \qquad c^1[1] \leftarrow \mathcal{E}(\mathsf{pk}[2], \mathsf{sk}[1]) \qquad c^1[0] \leftarrow \mathcal{E}(\mathsf{pk}[1], \mathsf{sk}[0])$$

to $\mathcal{A}$, which simulates $\mathsf{CIRC}_{\mathcal{A},\Pi}(3,\eta)$ when its internal coin is 1.
In the event $b=0$, $\mathcal{A}_1'$ has sent ciphertexts

$$\mathsf{s} \leftarrow \mathcal{E}(\mathsf{pk}[0],\mathsf{sk}[2]) \qquad c^0[1] \leftarrow \mathcal{E}(\mathsf{pk}[2],0^{|\mathsf{sk}[1]|}) \qquad c^0[0] \leftarrow \mathcal{E}(\mathsf{pk}[1],0^{|\mathsf{sk}[0]|})$$

to $\mathcal{A}$, which by Lemma 15, simulates $\mathsf{CIRC}_{\mathcal{A},\Pi}(3,\eta)$ when its internal coin is 0 except with negligible probability. Therefore we conclude that $\mathsf{CIRC}_{\mathcal{A}_1',\Pi'}(1,\eta) = \mathsf{CIRC}_{\mathcal{A},\Pi}(3,\eta)$ except with negligible probability.

To see Π' is 2-circular secure. Consider $(\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathcal{G}'(1^\eta), (\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathcal{G}'(1^\eta)$. Define adversary $\mathcal{A}_2'$ as follows:

$\mathcal{A}_2'$:

1. Receive η and $\mathsf{pk}_0 = \mathsf{pk}_0[0],\mathsf{pk}_0[1],\mathsf{pk}_0[2],\mathsf{s}_0$ and $\mathsf{pk}_1 = \mathsf{pk}_1[0],\mathsf{pk}_1[1],\mathsf{pk}_1[2],\mathsf{s}_1$. Send $\mathsf{pk}_1[0],\mathsf{pk}_1[2],\mathsf{pk}_0[1]$ to $\mathcal{A}$.
2. Receive $c_0^b = \langle 1, c_0^b[0], c_0^b[1]\rangle$ and $c_1^b = \langle 1, c_1^b[0], c_1^b[1]\rangle$. Send $\mathsf{s}_1, c_1^b[1], c_0^b[0]$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs b', output b'.

In the event $b=1$, $\mathcal{A}_2'$ has sent ciphertexts

$$\mathsf{s}_1 \leftarrow \mathcal{E}(\mathsf{pk}_1[0],\mathsf{sk}_1[2]) \qquad c_1^1[1] \leftarrow \mathcal{E}(\mathsf{pk}_1[2],\mathsf{sk}_0[1]) \qquad c_0^1[0] \leftarrow \mathcal{E}(\mathsf{pk}_0[1],\mathsf{sk}_1[0])$$

to $\mathcal{A}$, which simulates $\mathsf{CIRC}_{\mathcal{A},\Pi}(3,\eta)$ when its internal coin is 1.
In the event $b=0$, $\mathcal{A}_2'$ has sent ciphertexts

$$\mathsf{s}_1 \leftarrow \mathcal{E}(\mathsf{pk}_1[0],\mathsf{sk}_1[2]) \qquad c_1^0[1] \leftarrow \mathcal{E}(\mathsf{pk}_1[2],0^{|\mathsf{sk}_0[1]|}) \qquad c_0^0[0] \leftarrow \mathcal{E}(\mathsf{pk}_0[1],0^{|\mathsf{sk}_1[0]|})$$

By Lemma 15 we conclude that $\mathsf{CIRC}_{\mathcal{A}_2',\Pi'}(2,\eta) = \mathsf{CIRC}_{\mathcal{A},\Pi}(3,\eta)$ except with negligible probability.