An extended abstract of this paper appears in Marc Fischlin (Ed.): 15th IACR International Conference on Practice and Theory of Public-Key Cryptography - PKC 2012, Lecture Notes in Computer Science, Springer-Verlag, 2012. This is the full version.

New Definitions and Separations for Circular Security

David Cash* Matthew Green† Susan Hohenberger‡

Abstract

Traditional definitions of encryption security guarantee secrecy for any plaintext that can be computed by an outside adversary. In some settings, such as anonymous credential or disk encryption systems, this is not enough, because these applications encrypt messages that depend on the secret key. A natural question to ask is do standard definitions capture these scenarios? One area of interest is *n-circular security* where the ciphertexts $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ must be indistinguishable from encryptions of zero. Acar et al. (Eurocrypt 2010) provided a CPA-secure public key cryptosystem that is not 2-circular secure due to a distinguishing attack.

In this work, we consider a natural relaxation of this definition. Informally, a cryptosystem is *n-weak circular secure* if an adversary given the cycle $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ has no significant advantage in the regular security game, (e.g., CPA or CCA) where ciphertexts of chosen messages must be distinguished from ciphertexts of zero. Since this definition is sufficient for some practical applications and the Acar et al. counterexample no longer applies, the hope is that it would be easier to realize, or perhaps even implied by standard definitions. We show that this is unfortunately not the case: even this weaker notion is not implied by standard definitions. Specifically, we show:

- For symmetric encryption, under the minimal assumption that one-way functions exist, n-weak circular (CPA) security is not implied by CCA security, for any n. In fact, it is not even implied by authenticated encryption security, where ciphertext integrity is guaranteed.
- For public-key encryption, under a number-theoretic assumption, 2-weak circular security is not implied by CCA security.

In both of these results, which also apply to the stronger circular security definition, *we actually show for the first time an attack in which the adversary can recover the secret key of an otherwise-secure encryption scheme after an encrypted key cycle is published.* These negative results are an important step in answering deep questions about which attacks are prevented by commonly-used definitions and systems of encryption. They say to practitioners: if key cycles may arise in your system, then even if you use CCA-secure encryption, your system may break catastrophically; that is, a passive adversary might be able to recover your secret keys.

Keywords: Encryption, Definitions, Circular Security, Counterexamples

1 Introduction

Encryption is one of the most fundamental cryptographic primitives. Most definitions of encryption security [21, 18, 34] follow the seminal notion of Goldwasser and Micali which guarantees indistinguishability of

*IBM T.J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, N.Y. 10598, cdc@ucsd.edu. This work was performed at University of California, San Diego, supported in part by NSF grant CCF-0915675.

†Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218. Supported in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-11-2-0211, the Office of Naval Research under contract N00014-11-1-0470, NSF grant CNS-1010928 and HHS 90TR0003/01. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS mgreen@cs.jhu.edu

‡Johns Hopkins University. Supported in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-11-2-0211, the Office of Naval Research under contract N00014-11-1-0470, NSF CNS 1154035, a Microsoft Faculty Fellowship and a Google Faculty Research Award. Applying to all authors, the views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

encryptions for messages chosen by the adversary [21]. However, Goldwasser and Micali wisely warned to be careful when using a system proven secure within this framework on messages that the adversary cannot derive himself.

Over the past several years, there has been significant interest in designing schemes secure against *key-dependent message attacks*, e.g., [15, 11, 30, 3, 26, 28, 13, 14, 5, 2], where the system must remain secure even when the adversary is allowed to obtain encryptions of messages that depend on the secret keys themselves. In this work, we are particularly interested in circular security [15]. A public-key cryptosystem is *n-circular secure* if the ciphertexts $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$, as well as ciphertexts of chosen messages, cannot be distinguished from encryptions of zero, for independent key pairs. Either by design or accident, these key cycles naturally arise in many applications, including storage systems such as BitLocker [13], anonymous credentials [15], the study of "axiomatic security" [30, 3] and more. See [13] for a discussion of the applications.

Until recently, few positive or negative results regarding circular security were known outside of the random oracle model. On one hand, no n-circular secure cryptosystems were known for $n > 1$. On the other hand, no counterexamples existed for $n > 1$ to separate the definitions of circular and CPA security; that is, as far as anyone knew the CPA-security definition already captured circular security for any cycle larger than a self-loop.

Recently, this gap has been closing in two ways. On the positive side, several circular-secure schemes have been proposed [13, 5, 14]. The focus of the current work is on negative results – namely, investigating whether standard notions of encryption are "safe" for circular applications.

In 2008, Boneh, Halevi, Hamburg and Ostrovsky proved, by counterexample, that *one-way* security does not imply circular security [13]. Recently, Acar, Beleniky, Bellare and Cash [2] proved that, under an assumption in bilinear groups, CPA-security does not imply circular security.

Our Results We narrow this gap even further by studying the extent to which standard definitions (e.g., CPA, CCA) imply a *weak* form of circular security. Our results are primarily negative.

1. Relaxing the Circular Security Notion. Perhaps the current formulation of circular security is "too strong"; that is, perhaps there is a relaxed notion of this definition which simultaneously satisfies many practical applications and yet is also *already* captured by standard security notions. This is an area worth investigating. We begin by proposing a natural relaxation called *weak circular security* where the adversary is handed an encrypted cycle $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ along with the public keys and then proceeds to play the CPA or CCA security game as normal (where these ciphertexts are also off-limits for the decryption oracle). We stress here that the encrypted cycle is *always* generated as described, and is never changed to encryptions of zero. This definition is intriguing, and perhaps of independent interest, for two reasons.

First, the Acar et al. [2] counterexample does *not* apply to it. That construction uses the bilinear map to test whether a sequence of ciphertexts contain a cycle or zeros. Here the adversary knows he's getting an encrypted cycle, but then must extract some knowledge from this that helps him distinguish two messages of his choosing.

Second, this definition appears sufficient for some practical settings. Using a weak circular secure encryption scheme, Alice and Bob could exchange keys with each other over an insecure channel knowing that: (1) Eve can detect that they did so, but (2) Eve cannot learn anything about their other messages. Similarly, an adversary scanning over a user's BitLocker storage may detect that her drive contains an encrypted cycle, but cannot read anything on her drive. In an anonymous credential system of Camenisch and Lysyanskaya [15], a user has multiple keys. To participate in the system, the user must encrypt them in a cycle, provide this cycle to the other users, and prove that she has done this correctly. Then, if she shares one key, she automatically shares all her keys. In their application, *detection* of a cycle is actually desirable, provided that subsequent encryptions remain secure.

2. Symmetric-Key Counterexamples. In the symmetric setting, we show that standard notions do not imply n-circular security for any positive n. Specifically, given any $n \geq 1$, we show how to construct a

secure authenticated encryption scheme (which is necessarily CCA-secure; see Section 2) that is not n-weak circular secure, under the minimal assumption that secure authenticated encryption schemes exist, which are equivalent to one-way functions.

The main technical ingredient in our counterexample is a lemma showing that it is provably hard for an adversary to compute an encrypted key cycle itself, assuming that the symmetric scheme under attack is a secure authenticated encryption scheme (or CCA secure). We stress that this lemma does not hold if the encryption scheme is only CPA secure.

Our lemma gives us leverage in constructing a counterexample because it means the adversary is given strictly more power in the weak circular security game than in the standard security game. Specifically, the adversary is given an encrypted key cycle in the weak circular security game that it could not have computed itself, and we design a scheme to help such an adversary without affecting regular security.

3. Public-Key Counterexamples. We show that neither CPA nor CCA-security imply (even) weak circular security for cycles of size 2. That is, we show secure systems that are totally compromised when the independently-generated ciphertexts $E(pk_A, sk_B)$ and $E(pk_B, sk_A)$ are released. This is a difficult task, because the system must remain secure if either one, but only one, of these ciphertexts are released. Moreover, this counterexample requires new ideas. We cannot use the common trick in self-loop counterexamples that test if the message is the secret key corresponding to the public key, since there is no way for the encryption algorithm with public key pk_A to distinguish, say, sk_B from any other valid message. Specifically, we show that:

If there exists an algebraic setting where the Symmetric External Diffie-Hellman [1] (SXDH) assumption holds, then there exists a CPA-secure cryptosystem which is *not* 2-weak circular secure. The proposed scheme is particularly interesting in that it breaks *catastrophically* in the presence of a 2-cycle — revealing the secret keys of both users.

Moreover, if simulation-sound non-interactive zero- knowledge (NIZK) proof systems exist for NP and there exists an algebraic setting where the Symmetric External Diffie-Hellman (SXDH) assumption holds, then there exists a CCA-secure cryptosystem which is *not* 2-weak circular secure. This is also the first separation of CCA security and (regular) circular security.

These results deepen our understanding of how to define "secure" encryption and which practical attacks are captured by the standard definitions. They also provide additional justification for the ongoing effort, e.g. [13, 14, 5], to develop cryptosystems which are provably circular secure.

1.1 Related Work

In 2001, Camenisch and Lysyanskaya [15] introduced the notion of *circular security* and used it in their anonymous credential system to discourage users from delegating their secret keys. They also showed how to construct a circular-secure cryptosystem from any CPA-secure cryptosystem in the random oracle model. Independently, Abadi and Rogaway [1] and Black, Rogaway, Shrimpton [11] introduced the more general notion of *key-dependent message* (KDM) security, where the encrypted messages might depend on an arbitrary function of the secret keys. Black et al. showed how to realize this notion in the random oracle model.

Halevi and Krawczyk [26] extended the work of Black et al. to look at KDM security for deterministic secret-key functions such as pseudorandom functions (PRFs), tweakable blockciphers, and more. They give both positive and negative results, including some KDM-secure constructions in the standard model for PRFs. In the symmetric setting, Hofheinz and Unruh [28] showed how to construct circular-secure cryptosystems in the standard model under relaxed notions of security. Backes, Pfitzmann and Scedrov [7] presented stronger notions of KDM security (some in the random oracle model) and discussed the relationships among these notions.

[1] The SXDH assumption states that there is a bilinear setting $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where the Decisional Diffie-Hellman (DDH) assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$. It has been extensively studied and used e.g., [20, 38, 31, 12, 8, 6, 23, 9, 24], perhaps most notably as a setting of the Groth-Sahai NIZK proof system [24].

In the public-key setting, Boneh, Halevi, Hamburg and Ostrovsky [13] presented the first cryptosystem which is simultaneously CPA-secure and n-circular-secure (for any n) in the standard model, based on either the DDH or Decision Linear assumptions. As mentioned earlier, Boneh et al. [13] also proved, by counterexample, that *one-way* security does not imply circular security. One-way encryption is a very weak notion, which informally states that given $(pk, E(pk, m))$, the adversary should not be able to recover m. Given any one-way encryption system, they constructed a one-way encryption system that is not n-circular secure (for any n). Their system generates two key pairs from the original and sets $PK = pk_1$ and $SK = (sk_1, sk_2)$. A message (m_1, m_2) is encrypted as $(m_1, E(pk_1, m_2))$. In the event of a 2-cycle, the values $\mathsf{Enc}(pk_A, sk_B) = (sk_{B,1}, E(pk_{A,1}, sk_{B,2}))$ and $\mathsf{Enc}(pk_B, sk_A) = (sk_{A,1}, E(pk_{B,1}, sk_{A,2}))$ provide the critical secret key information $(sk_{B,1}, sk_{A,1})$ in the clear.

Subsequently, Applebaum, Cash, Peikert and Sahai [5] adapted the circular-secure construction of [13] into the lattice setting. Camenisch, Chandran and Shoup [14] extended[13] to the first cryptosystem which is simultaneously CCA-secure and n-circular-secure (for any n) in the standard model, by applying the "double encryption" paradigm of Naor and Yung [33]. (Interestingly, we use this same approach in Section 4.4 to extend our public-key counterexample from CPA to CCA security.)

Haitner and Holenstein [25] recently provided strong impossibility results for KDM-security *with respect to 1-key cycles* (a.k.a., self-loops.) They study the problem of building an encryption scheme where it is secure to release $E(k, g(k))$ for various functions g. First, they show that there exists no fully-black-box reduction from a KDM-secure encryption scheme to one-way permutations (or even some families of trapdoor permutations) if the adversary can obtain encryptions of $g(k)$, where g is a poly(n)-wise independent hash function. Second, there exists no reduction from an encryption scheme secure against key-dependent messages to, essentially, any cryptographic assumption, if the adversary can obtain an encryption of $g(k)$ for an *arbitrary* g, as long as the security reduction treats both the adversary and the function g as black boxes. These results address the possibility of achieving strong single-user KDM-security via reductions to cryptographic assumptions. The results in this paper study a version of KDM security that is in one sense weaker – we only allow a narrow class of functions g – but also stronger because it considers multiple users. Our results also address a different question regarding KDM security. We study whether or not KDM security is always implied by regular security while Haitner and Holenstein study the possibility of achieving strong single-user KDM security via specialized constructions.

Most closely related to our work, Acar et al. [2] demonstrated both public and private key encryption systems that are provably CPA-secure and yet also demonstrably *not* 2-circular secure. Their counterexample does not apply to CCA or weak circular security.

Subsequent to the original posting of this work, Rothblum [36] studied the circular security of bit encryption. In particular, using n-linear maps, for large n, where DDH is assumed hard in every pre-image group, he constructs a CPA (or CCA) secure bit-encryption scheme that is not circular secure; that is, where it is not "safe" to encrypt the secret key sk bit-by-bit using the corresponding public key pk. This approach is conceptually similar to extending either the Acar et al. [2] or our 2-circular counterexample in Section 4 to an n-circular counterexample using n-linear maps. Unfortunately, there are no candidate implementations for n-linear maps where $n > 2$ and even the discrete logarithm problem is believed to be hard in one of the pre-image groups. Thus, it remains an open problem to resolve these two fascinating questions relating to circular security.

There is also a relationship to recent work on *leakage resilient* and *auxiliary input* models of encryption, which mostly falls into the "self-loop" category. In leakage resilient models, such as those of Akavia, Goldwasser and Vaikuntanathan [4] and Naor and Segev [32], the adversary is given some function h of the secret key, not necessarily an encryption, such that it is *information theoretically* impossible to recover sk. The auxiliary input model, introduced by Dodis, Kalai and Lovett [17], relaxes this requirement so that it only needs to be difficult to recover sk.

Self-Loops In sharp contrast to all $n \geq 2$, the case of 1-circular security is fairly well understood. A folklore counterexample shows that CPA-security does not directly imply 1-circular security. Given any encryption scheme (G, E, D), one can build a second scheme (G, E', D') as follows: (1) $E'(pk, m)$ outputs

$\underline{\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(m_0, m_1, z) \leftarrow \mathcal{A}_1(pk)$
$y \leftarrow \mathsf{Enc}(pk, m_b)$
$\hat{b} \leftarrow \mathcal{A}_2(y, z)$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{AE}(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
$K \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\hat{b} \leftarrow \mathcal{A}^{\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot), \mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)}(1^\lambda)$
Output $(\hat{b} \stackrel{?}{=} b)$.

Figure 1: Experiments for Definitions 2.1 and 2.3.

$E(pk, m)||0$ if $m \neq sk$ and $m||1$ otherwise, (2) $D'(sk, c||b)$ outputs $D(sk, m)$ if $b = 0$ and sk otherwise. It is easy to show that if (G, E, D) is CPA-secure, then (G, E', D') is CPA-secure. When $E'(pk, sk) = sk||1$ is exposed, then there is a complete break. Conversely, given any CPA-secure system, one can build a 1-circular secure scheme in the standard model [13].

2 Definitions of Security

A *public-key encryption system* Π is a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, where KeyGen is a key-generation algorithm that takes as input a security parameter λ and outputs a public/secret key pair (pk, sk); $\mathsf{Enc}(pk, m)$ encrypts a message m under public key pk; and $\mathsf{Dec}(sk, c)$ decrypts ciphertext c with secret key sk. A *symmetric-key encryption system* is a public-key encryption system, except that it always outputs $pk = \perp$, and the encryption algorithm computes ciphertexts using sk, i.e. by running $\mathsf{Enc}(sk, m)$. In the symmetric case we will sometimes write K instead of sk. As in most other works, we assume that all algorithms implicitly have access to shared public parameters establishing a common algebraic setting.

Our definitions of security will associate a message space, denoted M, with each encryption scheme. Throughout this paper, we assume that the space of possible secret keys output by KeyGen is a subset of the message space M and thus any secret key can be encrypted using any public key. For symmetric encryption schemes we will always have $M \subset \{0,1\}^*$.

By $\nu(k)$ we denote some *negligible* function, i.e., one such that, for all $c > 0$ and all sufficiently large k, $\nu(k) < 1/k^c$. We abbreviate probabilistic polynomial time as PPT.

2.1 Standard Security Definitions

Public-key encryption We recall the standard notion of indistinguishability of encryptions under a chosen-plaintext attack due to Goldwasser and Micali [21].

Definition 2.1 ($\mathsf{IND\text{-}CPA}$) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme for the message space* M*. For* $b \in \{0,1\}$*,* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *and* $\lambda \in \mathbb{N}$*, let the random variable* $\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)$ *be defined by the probabilistic algorithm described on the left side of Figure 1. We denote the* $\mathsf{IND\text{-}CPA}$ advantage of $\mathcal{A}$ *by* $\mathsf{Adv}^{\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda) = 1] - 1$*. We say that* Π *is* $\mathsf{IND\text{-}CPA}$ secure *if* $\mathsf{Adv}^{\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

We also consider the indistinguishability of encryptions under chosen-ciphertext attacks [33, 34, 18].

Definition 2.2 ($\mathsf{IND\text{-}CCA}$) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme for the message space* M*. Let the random variable* $\mathsf{IND\text{-}CCA}(\Pi, \mathcal{A}, \lambda)$ *be defined by an algorithm identical to* $\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)$ *above, except that both* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *have access to an oracle* $\mathsf{Dec}(sk, \cdot)$ *that returns the output of the decryption algorithm and* $\mathcal{A}_2$ *cannot query this oracle on input* y*. We denote the* $\mathsf{IND\text{-}CCA}$ advantage of $\mathcal{A}$ *by* $\mathsf{Adv}^{\mathsf{cca}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CCA}(\Pi, \mathcal{A}, \lambda) = 1] - 1$*. We say that* Π *is* $\mathsf{IND\text{-}CCA}$ secure *if* $\mathsf{Adv}^{\mathsf{cca}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

$\underline{\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ to n:
  $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
If $b = 1$ then
  $\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})$
Else
  $\mathbf{y} \leftarrow \mathsf{EncZero}(\mathbf{pk}, \mathbf{sk})$
$\hat{b} \leftarrow \mathcal{A}(\mathbf{pk}, \mathbf{y})$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ to n:
  $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})$
$(j, m_0, m_1, z) \leftarrow \mathcal{A}_1(\mathbf{pk}, \mathbf{y})$
$y \leftarrow \mathsf{Enc}(pk_j, m_b)$
$\hat{b} \leftarrow \mathcal{A}_2(y, z)$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})}$
For $i = 1$ to n
  $y_i \leftarrow \mathsf{Enc}(pk_i, sk_{(i \bmod n)+1})$
Output $\mathbf{y}$

$\underline{\mathsf{EncZero}(\mathbf{pk}, \mathbf{sk})}$
For $i = 1$ to n
  $y_i \leftarrow \mathsf{Enc}(pk_i, 0^{|sk_{(i \bmod n)+1}|})$
Output $\mathbf{y}$

Figure 2: Experiments for Definitions 2.4 and 2.5. Each is defined with respect to a message space M, and we assume that $m_0, m_1 \in M$ always. We write $\mathbf{pk}$, $\mathbf{sk}$, and $\mathbf{y}$ for $(pk_1, \ldots, pk_n)$, $(sk_1, \ldots, sk_n)$ and $(y_1, \ldots, y_n)$ respectively.

Symmetric-key authenticated encryption We recall the definition of secure authenticated (symmetric-key) encryption due to [35], except that we will not require pseudorandom ciphertexts. Bellare and Namprempre [10] showed that AE implies $\mathsf{IND\text{-}CCA}$, and is in fact strictly stronger. For our counterexample, we target this very strong definition of security in order strengthen our results by showing that even this does not imply weak circular security.

Definition 2.3 (AE) *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme for the message space M. Let the random variable $\mathsf{AE}(\Pi, \mathcal{A}, \lambda)$ be defined by the probabilistic algorithm described on the right side of Figure 1. In the experiment, the oracle $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$ takes as input a pair of equal-length messages (m_0, m_1) and computes $\mathsf{Enc}(K, m_b)$. The oracle $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ takes as input a ciphertext c and computes $\mathsf{Dec}(K, c)$ if $b = 1$ and always returns $\perp$ if $b = 0$. The adversary is not allowed to submit any ciphertext to $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ that was previously returned by $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$. We denote the AE advantage of $\mathcal{A}$ by $\mathsf{Adv}^{\mathsf{ae}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{AE}(\Pi, \mathcal{A}, \lambda) = 1] - 1$. We say that Π is AE secure if $\mathsf{Adv}^{\mathsf{ae}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.*

2.2 Circular Security Definitions

We next give definitions for circular security of public-key and symmetric-key encryption. These definitions are variants of the Key-Dependent Message (KDM) security notion of Black et al. [11]. By restricting the adversary's power, we make it significantly harder for us to devise a counterexample and thus prove a stronger negative result.[2]

Definition 2.4 ($\mathsf{IND\text{-}CIRC\text{-}CPA}^n$) *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme for the message space M. For $b \in \{0,1\}$, integer $n > 0$, adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$, let the random variable $\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ be defined by the probabilistic algorithm on the left side of Figure 2. We denote the $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ advantage of $\mathcal{A}$ by*

$$\mathsf{Adv}^{n\text{-}\mathsf{circ}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π is $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ secure if $\mathsf{Adv}^{n\text{-}\mathsf{circ}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.

One could augment this definition by modifying the $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ experiment to allow for a challenge "left-or-right" query as in $\mathsf{IND\text{-}CPA}$. While this is a quite natural modification, it only strengthens the definition, and we are interested in studying the weakest notions for which we can give a separation. Next we give a definition of *weak* circular security of public-key encryption.

[2] If we allowed the adversary to obtain encryptions of any affine function of the secret keys, as is done in [26, 13], then we could devise a trivial counterexample where the adversary uses 1-cycles to break the system.

Definition 2.5 (IND-WCIRC-CPA^n) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme for the message space* M*. For* $b \in \{0,1\}$*, integer* $n > 0$*, adversary* $\mathcal{A}$ *and* $\lambda \in \mathbb{N}$*, let the random variable* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ *be defined by probabilistic algorithm on the center of Figure 2. We denote the* IND-WCIRC-CPA^n advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π *is* IND-WCIRC-CPA^n secure *if the function* $\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

Finally, we give a definition of weak circular security for symmetric encryption. We will abuse notation and also call this IND-WCIRC-CPA^n security, since it will be clear from the context whether or not we mean public-key and symmetric-key.

Definition 2.6 (IND-WCIRC-CPA^n) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a symmetric-key encryption scheme for the message space* M*. For* $b \in \{0,1\}$*, integer* $n > 0$*, adversary* $\mathcal{A}$ *and* $\lambda \in \mathbb{N}$*, let* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ *be defined by the following probabilistic algorithm:*

$\underline{\mathsf{IND\text{-}WCIRC\text{-}CPA}^n_b(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ *to* n*:*
  $K_i \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{K})$
$\hat{b} \leftarrow \mathcal{A}^{\widetilde{\mathsf{Enc}}(\cdot,\cdot,\cdot)}(\mathbf{y})$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{EncCycle}(\mathbf{K})}$
For $i = 1$ *to* n
  $y_i \leftarrow \mathsf{Enc}(K_i, K_{(i \bmod n)+1})$
Output $\mathbf{y}$

$\underline{\widetilde{\mathsf{Enc}}(j, m_0, m_1)}$
Return $\mathsf{Enc}(K_j, m_b)$

We denote the IND-WCIRC-CPA^n advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π *is* IND-WCIRC-CPA^n secure *if* $\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

Discussion In both the IND-CPA and IND-CIRC-CPA notions, the adversary must distinguish an encryption (or encryptions) of a special message from the encryption of zero. This choice of the message zero is arbitrary. We keep it in the statement of our definition to be consistent with [13]; however, it is important to note, for systems such as ours where zero is not in the message space, that zero can be replaced by any constant message for an equivalent definition. Acar et al. [2] use an equivalent definition where zero is replaced by a fresh random message.

We will not need to define a notion of security to withstand *circular and chosen-ciphertext attacks*, because we are able to show a stronger negative result. In Section 4.4, we provide an IND-CCA-secure cryptosystem, which is provably not IND-CIRC-CPA-secure. In other words, we are able to devise a peculiar cryptosystem: one that withstands all chosen-ciphertext attacks, and yet breaks under a weak circular attack which does not require a decryption oracle.

3 Counterexample for Symmetric Encryption

Encryption Scheme Π_{ae} Let $\Pi'_{\mathsf{ae}} = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ be a secure authenticated encryption scheme. To simplify our results, we assume that $\mathsf{KeyGen}'(1^\lambda)$ outputs a uniformly random key K in $\{0,1\}^\lambda$, that the message space $M' = \{0,1\}^*$, and that ciphertexts output by $\mathsf{Enc}'(K, m)$ are always in $\{0,1\}^{p(|m|)}$, where p is some polynomial that depends on λ. We also assume that the first λ bits of a ciphertext are *never* equal to K. All of these assumptions can be removed via straightforward and standard modifications to our arguments below.

Fix a positive integer n. We now construct our counterexample scheme, denoted $\Pi_{\mathsf{ae}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. We will take $\mathsf{KeyGen} = \mathsf{KeyGen}'$, i.e., Π_{ae} also uses keys randomly chosen from $\{0,1\}^\lambda$. The message-space of Π_{ae} will consist of $M = \{0,1\}^\lambda \cup \{0,1\}^{np(\lambda)}$, bit strings of length either λ or $np(\lambda)$. The algorithms Enc and Dec are defined as follows.

$\underline{\mathsf{Enc}(K, m)}$
  If $\mathsf{IsCycle}(K, m)$ then
    Output $K \parallel m$
  Else
    Output $\mathsf{Enc}'(K, m)$

$\underline{\mathsf{Dec}(K, c)}$
  If $c = K \parallel \tilde{m}$ then
    Output $\tilde{m}$
  Else
    Output $\mathsf{Dec}'(K, c)$

$\underline{\mathsf{IsCycle}(K, m)}$
  If $|m| \neq np(\lambda)$
    Return false
  Parse m as $(c_1, \ldots, c_n)$
  $K_2 \leftarrow \mathsf{Dec}'(K, c_1)$
  For $i = 2$ to n
    $K_{i \bmod n+1} \leftarrow \mathsf{Dec}'(K_i, c_i)$
  Return $(K_1 \stackrel{?}{=} K)$

Decryption is always correct. This follows from our assumption that Enc' will never output a ciphertext that contains K as a prefix. We first establish the AE security of our scheme.

Theorem 3.1 *Encryption scheme Π_{ae} is AE secure whenever Π'_{ae} is AE secure.*
(Proof in Appendix A.2.)

The proof proceeds by showing that computing an encrypted key-cycle during the AE game is equivalent to recovering the secret key. From there we can reduce the security of Π_{ae} to Π'_{ae} easily.

Curiously, Theorem 3.1 is no longer true if one replaces AE security with a symmetric version of $\mathsf{IND\text{-}CPA}$ security for both Π_{ae} and Π'_{ae}. Namely, some type of chosen-ciphertext security is required on Π'_{ae} to prove even chosen-plaintext security of Π_{ae}. Intuitively, this is because it might be possible for an adversary to compute an encrypted key-cycle on its own if the scheme is only $\mathsf{IND\text{-}CPA}$-secure, but *not* if the scheme is AE-secure. In fact, the work of Boneh et al. [13] gives an explicit example of a scheme where the adversary can compute a cycle himself.

The Attack We now show that Π_{ae} is not circular-secure for n cycles, even in a weak sense.

Theorem 3.2 *Π_{ae} is not $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ secure.*

Proof. We give an explicit adversary $\mathcal{A}$ that has advantage negligibly close to 1. The adversary takes as input the encrypted key-cycle $\mathbf{y}$ in the $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ game. It queries $\widetilde{\mathsf{Enc}}(1, m_0, m_1)$, where $m_0 = \mathbf{y}$ and m_1 is a random message of the same length. Let y be the ciphertext returned by the oracle.

At this point, there are many ways to proceed; perhaps the simplest is to observe that the *length* of y depends on the challenge bit b. This is because, if $b = 0$, then $m_0 = \mathbf{y}$ was encrypted, resulting in $y = K \parallel \mathbf{y}$, which is $\lambda + np(\lambda)$ bits long. If $b = 1$ then y was computed by running $\mathsf{Enc}'(K, m_1)$, which will be $p(|m_1|) = p(np(\lambda))$ bits long *if* $\mathsf{IsCycle}(K, m_1)$ *returns* false. Thus, as long as $\mathsf{IsCycle}(K, m_1)$ returns false, $\mathcal{A}_2$ can compute the value of b by measuring y's length.

But why should $\mathsf{IsCycle}(K, m_1)$ return false? This follows from the AE security of Π'_{ae}. Let us parse m_1 into $(c_1, \ldots, c_n)$, where each $c_i \in \{0,1\}^{p(\lambda)}$ is random. When $\mathsf{IsCycle}(K, m_1)$ returns true, it must be that $\mathsf{Dec}'(K, c_1)$ did not return $\perp$. But if this happens, then we can construct an adversary to break the AE security of Π'_{ae}. The adversary simply queries $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ at a random point, observes if it returns $\perp$ or not, and outputs $\hat{b} = 0$ or 1 depending on this observation. □

We note that we could design an encryption scheme that does not have this type of ciphertext-length behavior by giving a different attack that abuses the fact that K is present in the ciphertext in one case, but not the other. We have chosen to present the attack this way for simplicity only.

4 Counterexamples for Public-Key Encryption

4.1 Preliminaries and Algebraic Setting

Bilinear Groups We work in a bilinear setting where there exists an efficient mapping function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ involving groups of the same prime order p. Two algebraic properties required are that: (1) if g generates $\mathbb{G}_1$ and h generates $\mathbb{G}_2$, then $e(g,h) \neq 1$ and (2) for all $a, b \in \mathbb{Z}_p$, it holds that $e(g^a, h^b) = e(g,h)^{ab}$.

Decisional Diffie-Hellman Assumption (DDH) Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^\lambda)$. For all PPT adversaries $\mathcal{A}$, the following probability is $1/2$ plus an amount negligible in λ:

$$\Pr\left[\begin{array}{l} g, z_0 \leftarrow \mathbb{G}; a, b \leftarrow \mathbb{Z}_p; z_1 \leftarrow g^{ab}; d \leftarrow \{0,1\}; \\ d' \leftarrow \mathcal{A}(g, g^a, g^b, z_d) : d = d' \end{array}\right].$$

Strong External Diffie-Hellman Assumption (SXDH): Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be bilinear groups. The SXDH assumption states that the DDH problem is hard in both $\mathbb{G}_1$ and in $\mathbb{G}_2$. This implies that there does *not* exist an efficiently computable isomorphism between these two groups. The SXDH assumption appears in many prior works, such as [20, 38, 31, 12, 8, 6, 23, 9, 24, 2].

Indistinguishability and Pseudorandom Generators

Definition 4.1 (Indistinguishability) *Two ensembles of probability distributions $\{X_k\}_{k\in\mathbb{N}}$ and $\{Y_k\}_{k\in\mathbb{N}}$ with index set $\mathbb{N}$ are said to be* computationally indistinguishable *if for every polynomial-size circuit family $\{D_k\}_{k\in\mathbb{N}}$, there exists a negligible function ν such that*

$$|\Pr[x \leftarrow X_k : D_k(x) = 1] - \Pr[y \leftarrow Y_k : D_k(y) = 1]|$$

is less than $\nu(k)$. We denote such sets $\{X_k\}_{k\in\mathbb{N}} \stackrel{c}{\approx} \{Y_k\}_{k\in\mathbb{N}}$.

Definition 4.2 (Pseudorandom Generator [29]) *Let U_x denote the uniform distribution over $\{0,1\}^x$. Let $\ell(\cdot)$ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any input $s \in \{0,1\}^n$, algorithm G outputs a string of length $\ell(n)$. We say that G is a* pseudorandom generator *if the following two conditions hold:*

- *(Expansion:) For every n, it holds that $\ell(n) > n$.*
- *(Pseudorandomness:) For every n, $\{U_{\ell(n)}\}_n \stackrel{c}{\approx} \{s \leftarrow U_n : G(s)\}_n$.*

The constructions of Section 4.2 use a PRG where the domain of the function is an exponentially-sized cyclic group.

4.2 Encryption Scheme Π_{cpa}

We now describe an encryption scheme $\Pi_{\mathsf{cpa}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. It is set in asymmetric bilinear groups $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order p where we assume that the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct and that the DDH assumption holds in both. We assume that a single set of group parameters $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h)$, where $\mathbb{G}_1 = \langle g \rangle, \mathbb{G}_2 = \langle h \rangle$, will be shared across all keys generated at a given security level and are implicitly provided to all algorithms.

The message space is $\mathcal{M} = \{0,1\} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$. Let $\mathsf{encode} : \mathcal{M} \to \{0,1\}^{\ell(\lambda)}$ and $\mathsf{decode} : \{0,1\}^{\ell(\lambda)} \to \mathcal{M}$ denote an invertible encoding scheme where $\ell(\lambda)$ is the polynomial length of the encoded message. Let $F : \mathbb{G}_T \to \{0,1\}^{\ell(\lambda)}$ be a pseudorandom generator secure under the Decisional Diffie Hellman assumption. (Recall that pseudorandom generators can be constructed from any one-way function [27].)

$\mathsf{KeyGen}(1^\lambda)$. The key generation algorithm selects a random bit $\beta \leftarrow \{0,1\}$ and random values $a_1, a_2 \leftarrow \mathbb{Z}_p^*$. The secret key is set as $sk = (\beta, a_1, a_2)$. We note that $sk \in \mathcal{M}$. The public key is set as:

$$pk = \begin{cases} (0, e(g,h)^{a_1}, g^{a_2}) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_1 & \text{if } \beta = 0 \\ (1, e(g,h)^{a_1}, h^{a_2}) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_2 & \text{if } \beta = 1. \end{cases}$$

$\mathsf{Encrypt}(pk, M)$. The encryption algorithm parses the public key $pk = (\beta, Y_1, Y_2)$, where Y_2 may be in $\mathbb{G}_1$ or $\mathbb{G}_2$ depending on the structure of the public key, and message $M = (\alpha, m_1, m_2) \in \mathcal{M}$. Note that m_1 and m_2 cannot be zero, but these values can be easily included in the message space by a proper encoding.

Select random $r \leftarrow \mathbb{Z}_p$ and $R \leftarrow \mathbb{G}_T$. Set $I = F(R) \oplus \mathsf{encode}(M)$.

Output the ciphertext C as:

$$C = \begin{cases} (g^r,\ R \cdot Y_1^r,\ Y_2^{rm_2} \cdot g^{m_1},\ I) & \text{if } \beta = 0; \\ (h^r,\ R \cdot Y_1^r,\ Y_2^{rm_2} \quad\quad ,\ I) & \text{if } \beta = 1. \end{cases}$$

We note that in the first case, $C \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)}$, while in the second $C \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)}$.

$\mathsf{Decrypt}(sk, C)$. The decryption algorithm parses the secret key $sk = (\beta, a_1, a_2)$ and the ciphertext $C = (C_1, C_2, C_3, C_4)$. Next, it computes:

$$R = \begin{cases} (C_2 / e(C_1, h))^{a_1} & \text{if } \beta = 0; \\ (C_2 / e(g, C_1))^{a_1} & \text{if } \beta = 1. \end{cases}$$

Then it computes $M' = F(R) \oplus C_4 \in \{0,1\}^{\ell(\lambda)}$ and outputs the message $M = \mathsf{decode}(M')$.

Discussion Like the circular-secure scheme of Boneh et al. [13], the above cryptosystem is a variation on El Gamal [19]. It is a practical system, which on first glance might be somewhat reminiscent of schemes the readers are used to seeing in the literature. The scheme includes a few "artificial" properties: (1) placing a public key in either $\mathbb{G}_1$ or $\mathbb{G}_2$ at random and (2) the fact that the ciphertext value C_3 is unused in the decryption algorithm. We will shortly see that these features are "harmless" in a semantic-security sense, but very useful for recovering the secret keys of the system in the presence of a two cycle. While it is not unusual for counterexamples to have artificial properties (e.g., [16, 22]), we can address these points as well.[3] In Appendix C, we show that property (1) can be removed by doubling the length of the ciphertext. For property (2), we observe that many complex protocols such as group signatures (e.g., [12]) combine ciphertexts with other components that are unused in decryption but are quite important to the protocol as a whole. Thus, we believe our counterexample is not that far fetched. It is possible that such an attack could exist on one of today's commonly-used encryption algorithms.

We first show that Π_{cpa} meets the standard notion of CPA security.

Theorem 4.3 *Encryption scheme* Π_{cpa} *is* IND-CPA *secure under the Decisional Diffie-Hellman Assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *(SXDH).*

The proof is given in Appendix B. It is relatively standard and involves repeated applications of the DDH assumption and PRG security.

[3] While our scheme is different from that of Acar et al. [2], that scheme also has similar artificial properties such as the presence of values that are not used in decryption.

4.3 The Attack

Despite being IND-CPA-secure, cryptosystem Π_{cpa} is not even weakly circular secure for 2-cycles. Specifically, given a circular encryption of two keys, we show that an adversary can distinguish another ciphertext with advantage 1/2. Our adversary actually does much more than this: with probability 1/2 over the coins used in key generation, *it can recover both secret keys.*

This is the first circular attack that allows the adversary to recover the secret keys. (In Appendix C, we discuss how to improve these probabilities to almost 1.) Our attack combines elements of both ciphertexts in an attempt to recover sk_A, which can then be used to decrypt the first ciphertext and obtain sk_B. It is counterintuitive that this is possible, given that it is easy to see that IND-CPA-security guarantees that it is safe for *one* of them to send their message.

Theorem 4.4 Π_{cpa} *is not* IND-WCIRC-CPA^2*-secure.*

Proof. We give PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathsf{Adv}^{\text{2-wcirc-cpa}}_{\Pi_{\mathsf{cpa}},\mathcal{A}}(\lambda)$ is equal to 1/2. Since IND-WCIRC-CPA security requires that this advantage be negligible, this attack breaks security. The adversary proceeds as follows. The first stage of the adversary, $\mathcal{A}_1$, obtains the two public keys, which we will write as pk_A and pk_B, and an encrypted cycle, which we will write as (C_A, C_B).

If both keys have $\beta = 0$ or $\beta = 1$ (call this event E_1), the adversary aborts and instructs the second stage ($\mathcal{A}_2$) to output a random bit. Since the two keys are independently generated by the challenger, this event will occur with probability exactly 1/2. Below we will condition on E_1 not happening, and wlog assume that $pk_A = (0, e(g,h)^{a_1}, g^{a_2})$ and $pk_B = (1, e(g,h)^{b_1}, h^{b_2})$. The corresponding secret keys $sk_A = (0, a_1, a_2)$, $sk_B = (1, b_1, b_2)$ are not known to the adversary.

We write the given ciphertexts $C_A = (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4})$ and $C_B = (c_{B,1},\ c_{B,2},\ c_{B,3},\ c_{B,4})$. $\mathcal{A}_1$ will output two arbitrary distinct messages, and request that the challenge use pk_A. For the state passed to $\mathcal{A}_2$, it now computes:

$$X := c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})}.$$

$\mathcal{A}_1$ sets $\hat{sk}_A = \mathsf{decode}(c_{B,4} \oplus F(X))$ and passes this with the challenge messages as state to $\mathcal{A}_2$.

$\mathcal{A}_2$ receives a ciphertext y and the passed state. It parses $\hat{sk}_A$ as a secret key for Π_{cpa} and computes $\mathsf{Dec}(\hat{sk}_A, y)$, and tests if this is equal to either of the challenge messages. If so, it outputs the corresponding bit. Otherwise it outputs a random bit.

Let's explore why this test works. Write $C_A = \mathsf{Enc}(pk_A, sk_B)$ and $C_B = \mathsf{Enc}(pk_B, sk_A)$. Then:

$$\begin{aligned}
C_A &= (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4}) \\
&= (g^r, R \cdot e(g,h)^{ra_1}, g^{ra_2b_2+b_1}, F(R) \oplus \mathsf{encode}(sk_B)) \\
C_B &= (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4}) \\
&= (h^s, S \cdot e(g,h)^{sb_1}, h^{sa_2b_2}, F(S) \oplus \mathsf{encode}(sk_A))
\end{aligned}$$

for some $r, s \in \mathbb{Z}_p$ and $R, S \in \mathbb{G}_T$. Then we have that:

$$\begin{aligned}
X := c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})} &= S \cdot e(g,h)^{sb_1} \cdot \frac{e(g^r, h^{sa_2b_2})}{e(g^{ra_2b_2+b_1}, h^s)} \\
&= S \cdot e(g,h)^{sb_1} \cdot \frac{e(g,h)^{rsa_2b_2}}{e(g,h)^{rsa_2b_2} \cdot e(g,h)^{sb_1}} = S.
\end{aligned}$$

Thus, $\mathcal{A}_1$ recovers $\hat{sk}_A = sk_A$ as $\mathsf{decode}(c_{B,4} \oplus F(S))$, and $\mathcal{A}_2$ will correctly guess bit b in this case.

Write $\hat{b}$ for the output of $\mathcal{A}_2$. We have

$$\begin{aligned}\mathsf{Adv}^{\text{2-wcirc-cpa}}_{\Pi_{\mathsf{cpa}},\mathcal{A}}(\lambda) &= 2\Pr[\hat{b} = b] - 1 \\ &= 2(\Pr[\hat{b} = b | E_1]\Pr[E_1] + \\ &\qquad \Pr[\hat{b} = b | \neg E_1]\Pr[\neg E_1]) - 1 \\ &= 2(1 \cdot 1/2 + 1/2 \cdot 1/2) - 1 \\ &= 1/2\end{aligned}$$

This completes the proof. □

4.4 Extension: A Counterexample for CCA Security

We show that there exists an IND-CCA-secure cryptosystem, which suffers a complete break when Alice and Bob trade secret keys over an insecure channel; i.e., transmit the two-key cycle $E(pk_A, sk_B)$ and $E(pk_B, sk_A)$. Our construction follows the "double-encryption" approach to building IND-CCA systems from IND-CPA systems as pioneered by Naor and Yung [33] and refined by Dolev, Dwork and Naor [18] and Sahai [37]. Our building blocks will be:

1. The IND-CPA-secure cryptosystem $\Pi_{\mathsf{cpa}} = (G, E, D)$ from Section 4. Let $E(pk, m; r)$ be the encryption of m under public key pk with randomness r.
2. An adaptively non-malleable non-interactive zero-knowledge (NIZK) proof system with unpredictable simulated proofs and uniquely applicable proofs for the language L of consistent pairs of encryptions, defined as:
$$L = \left\{ \begin{array}{l} (e_0, e_1, c_0, c_1) : \quad \exists m, r_0, r_1 \in \{0,1\}^* \text{ s.t.} \\ c_0 = E(e_0, m; r_0) \text{ and } c_1 = E(e_1, m; r_1) \end{array} \right\}.$$

A proof system for L can be realized under relatively mild assumptions, such as the difficulty of factoring Blum integers (e.g., [37]). One complication is that the secret keys for this cryptosystem now change and the construction must be adapted accordingly, so that the secret key can still be recovered by the adversary during a circular attack. We show that this is possible.

Construction Π_{cca}. The construction $\Pi_{\mathsf{cca}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, following [37] directly, is then defined as follows. Let $t(\lambda)$ be the polynomial bound on the amount of randomness needed by the encryption algorithm to encrypt a single message and let $q(\lambda)$ be the polynomial length of the reference string required by the proof system Γ.

$\mathsf{KeyGen}(1^\lambda)$. Call $G(1^\lambda)$ twice to generate two key pairs (e_0, d_0) and (e_1, d_1). Select a random reference string $\Sigma \in \{0,1\}^{q(\lambda)}$ for Γ. Set $pk = (e_0, e_1, \Sigma)$ and $sk = (d_0, d_1)$.

$\mathsf{Encrypt}(pk, M \in (\{0,1\} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*)^2)$. Choose random $r_0, r_1 \leftarrow \{0,1\}^{t(k)}$. Let $c_0 = E(e_0, m; r_0)$ and $c_1 = E(e_1, m; r_1)$. Use P to generate a proof π relative to Σ that $(e_0, e_1, c_0, c_1) \in L$ using (m, r_0, r_1) as the witness. Output the ciphertext (c_0, c_1, π).

$\mathsf{Decrypt}(sk, C)$. Use V to verify the correctness of π. If π is valid, output either of $D(d_0, c_0)$ or $D(d_1, c_1)$, chosen arbitrarily.

Theorem 4.5 *Encryption scheme Π_{cca} is IND-CCA secure under the Decisional Diffie-Hellman Assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$ (SXDH) and the assumption that proof system Γ satisfies the above constraints.* (Follows directly from Theorem 4.3 and [37], Theorem 4.1.)

Theorem 4.6 *Π_{cca} is not IND-WCIRC-CPA^2-secure.*

Proof sketch. Given two public keys $pk_A = (e_{A,0}, e_{A,1}, \Sigma_A)$ and $pk_B = (e_{B,0}, e_{B,1}, \Sigma_B)$, and two valid ciphertexts $C_A = (c_{A,0}, c_{A,1}, \pi_A)$ and $C_B = (c_{B,0}, c_{B,1}, \pi_B)$. The attack follows the same outline as that in the proof of Theorem 4.4, using the values $(e_{A,0}, e_{B,0}, c_{A,0}, c_{B,0})$ and ignoring the rest of the ciphertexts. If the encryption keys are of different types (not both type 0 or type 1), then the distinguisher will win with advantage 1/2 as before. □

5 Conclusion and Open Problems

In this work, we presented a natural relaxation of the circular security definition, which may prove interesting for positive results in its own right. We demonstrated that its guarantees are *not* already captured by standard definitions of encryption. To do this, we presented symmetric and public-key encryption systems that are secure in the IND-CPA and IND-CCA sense, but fail catastrophically in the presence of an encrypted cycle. This provides the first answer to the foundational question on whether IND-CCA-security captures (weak or regular) circular security for all cycles larger than self-loops. In either case, it does not.

Our work leaves open the interesting problem of finding a public-key counterexample for cycles of size ≥ 3. Secondly, while our symmetric counterexample depended only on the existence of AE-secure symmetric encryption, our public-key counterexample, like that of Acar et al. [2], required a specific bilinear map assumption. It would be highly interesting to find a counterexample assuming only that IND-CPA- or IND-CCA-secure systems exist.

Finally, we observe that our public-key counterexample contains a novel and curious property – *certain combinations of independently generated ciphertexts trigger the release of their underlying plaintext.* From Rabin's $\frac{1}{2}$-OT system to DH-DDH gap groups, the cryptographic community has a strong history of turning such oddities to an advantage. If we view a cryptosystem with this property as a new primitive, what new functionalities can be realized using it?

Acknowledgments

The authors thank Ronald Rivest for the suggestion to view the public key counterexample in Section 4 as a potential building block for other functionalities.

References

[1] Martin Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[2] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In *EUROCRYPT '10*, volume 6110 of LNCS, pages 403–422. Springer, 2010.

[3] Pedro Adao, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In *ESORICS '05*, volume 3679 of LNCS, pages 374–396, 2005.

[4] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC '09*, volume 5444 of LNCS, pages 474–495, 2009.

[5] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO '09*, volume 5677 of LNCS, pages 595–618, 2009.

[6] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *CCS '05*, pages 92–101, 2005.

[7] M. Backes, B. Pfitzmann, and A. Scedrov. Key-dependent message security under active attacks - BRSIM/UC-soundness of Dolev-Yao-style encryption with key cycles. *J.of Comp.Security*, 16(5):497–530, 2008.

[8] Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose. Correlation-resistant storage. Technical Report TR-SP-BGMM-050705, Johns Hopkins University, CS Dept, 2005. `http://spar.isi.jhu.edu/~mgreen/correlation.pdf`.

[9] Mira Belenkiy, Melissa Chase, Markulf Kolweiss, and Anna Lysyanskaya. Non-interactive anonymous credentials. In *TCC '08*, volume 4948 of LNCS, pages 356–374, 2008.

[10] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008.

[11] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC*, volume 2595 of LNCS, pages 62–75, 2002.

[12] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO '04*, volume 3152 of LNCS, pages 45–55, 2004.

[13] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-Secure Encryption from Decision Diffie-Hellman. In *CRYPTO '08*, volume 5157 of LNCS, pages 108–125, 2008.

[14] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT '09*, volume 5479 of LNCS, pages 351–368, 2009.

[15] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01*, volume 2045 of LNCS, pages 93–118, 2001.

[16] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. of the ACM*, 51(4):557–594, 2004.

[17] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC '09*, pages 621–630, 2009.

[18] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Computing*, 30(2):391–437, 2000.

[19] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO '84*, pages 10–18, 1984.

[20] Steven D. Galbraith. Supersingular curves in cryptography. In *ASIACRYPT '01*, volume 2248 of LNCS, pages 495–513, 2001.

[21] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[22] Shafi Goldwasser and Yael Tauman Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS '03*, page 102, 2003.

[23] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT*, volume 5350, pages 179–197, 2008.

[24] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT '08*, volume 4965 of LNCS, pages 415–432, 2008.

[25] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *TCC '09*, volume 5444 of LNCS, pages 202–219, 2009.

[26] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In *ACM CCS '07*, pages 466–475, 2007.

[27] Johan Hastad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Computing*, 28(4):1364–1396, 1999.

[28] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In *EUROCRYPT '08*, volume 4965 of LNCS, pages 108–126, 2008.

[29] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[30] Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In *ICISC*, volume 2971, pages 55–66, 2003.

[31] Noel McCullagh and Paulo S. L. M. Barreto. A new two-party identity-based authenticated key agreement. In *CT-RSA '04*, volume 3376, pages 262–274, 2004.

[32] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO '09*, volume 5677 of LNCS, pages 18–35, 2009.

[33] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC '90*, pages 427–437, 1990.

[34] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO '91*, volume 576 of LNCS, pages 433–444, 1991.

[35] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In *EUROCRYPT*, pages 373–390, 2006.

[36] Ron Rothblum. On the circular security of bit-encryption. Cryptology ePrint Archive, Report 2012/102, 2012. `http://eprint.iacr.org/`.

[37] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS '99*, pages 543–553, 1999.

[38] Mike Scott. Authenticated id-based key exchange and remote log-in with simple token and pin number, 2002. Available at `http://eprint.iacr.org/2002/164`.

A Security Proof for Π_{ae}

A.1 Security against Key Recovery Attacks

It will simplify our results to use the following concept of key recovery security, which is implied by AE security.

Definition A.1 (KR) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a symmetric-key encryption scheme for the message space* M*. Let the random variable* $\mathsf{KR}(\Pi, \mathcal{A}, \lambda)$ *be defined by the following probabilistic algorithm:*

$\underline{\mathsf{KR}(\Pi, \mathcal{A}, \lambda)}$
$K \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\hat{K} \leftarrow \mathcal{A}^{\mathcal{E}_K^{\mathrm{kr}}(\cdot), \mathcal{D}_K^{\mathrm{kr}}(\cdot)}(1^\lambda)$
Output $(\hat{K} \stackrel{?}{=} K)$.

Here the oracle $\mathcal{E}_K^{\mathrm{kr}}(\cdot)$ takes as input a message $m \in M$ and returns $\mathsf{Enc}(K, m)$, and the oracle $\mathcal{D}_K^{\mathrm{kr}}(\cdot)$ takes as input a ciphertext and returns $\mathsf{Dec}(K, c)$.

We denote the KR advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{\mathsf{kr}}_{\Pi,\mathcal{A}}(\lambda) = \Pr[\mathsf{KR}(\Pi, \mathcal{A}, \lambda) = 1].$$

We say that Π *is* KR secure *if* $\mathsf{Adv}^{\mathsf{kr}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$.

We will use the following theorem below. The proof is standard.

Theorem A.2 *Any* AE*-secure symmetric-key encryption scheme is also* KR*-secure.*

A.2 Proof of Security for System Π_{ae}

Theorem A.3 *Encryption scheme* Π_{ae} *is* AE *secure whenever* Π'_{ae} *is* AE *secure.*

Proof. We prove the theorem by giving a reduction to the AE security of Π'_{ae}. We proceed by describing a pair of hybrid games, where the first H_0 is defined to be the AE experiment from Definition 2.3 with Π_{ae}, and the second is a modified experiment that will be seen to be essentially equivalent to the AE experiment with Π'_{ae}.

We denote the hybrids $\mathsf{H}_0, \mathsf{H}_1$, and define them as follows:

H_0: The AE experiment with Π_{ae}.
H_1: Exactly as in H_0, except that the oracles $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$ and $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ use modified versions of the algorithms Enc and Dec which ignore their "If" statements and proceed directly the "Else" clause.

Fix some PPT adversary $\mathcal{A}$, and let

$$\mathsf{Adv}^{\mathsf{H}_i}_{\mathcal{A}}(\lambda) = 2\Pr[\mathsf{H}_i(\mathcal{A}, \lambda) = 1] - 1$$

for $i = 0, 1$. Then we have

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{ae}}_{\Pi_{\mathsf{ae}},\mathcal{A}}(\lambda), \tag{1}$$

which is negligible by assumption. Next we relate $\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda)$ and $\mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda)$.

Lemma A.4 *For all PPT adversaries* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) \leq \epsilon_1(\lambda) \tag{2}$$

for some negligible function ϵ_1.

Proof. Suppose to the contrary that a PPT adversary $\mathcal{A}$ exists that violates (2). Using $\mathcal{A}$ we construct an PPT adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{kr}}_{\Pi'_{\mathsf{ae}},\mathcal{B}}(\lambda)$ is non-negligible which contradicts the AE security Π'_{ae} by Theorem A.2.

The adversary $\mathcal{B}$ has access to two oracles in the KR experiment with Π'_{ae}, $\mathcal{E}_K^{\mathrm{kr}}(\cdot)$ and $\mathcal{D}_K^{\mathrm{kr}}(\cdot)$. $\mathcal{B}$ will run $\mathcal{A}$, which expects the two oracles $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot), \mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ in the AE experiment with Π_{ae}.

$\mathcal{B}$ starts by selecting $b \xleftarrow{r} \{0,1\}$ and initializing a list L to be empty. $\mathcal{B}$ then runs $\mathcal{A}$, simulating queries to $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$ and $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ as follows:

$\underline{\mathcal{E}^{\mathrm{ae}}_{K,b}(m_0, m_1)}$
  $\underline{\mathsf{UseCycle}(m_b)}$
  Return $\mathcal{E}_K^{\mathrm{kr}}(m_b)$

$\underline{\mathsf{UseCycle}(x)}$
  If $|x| \neq np(\lambda)$ then
    Return
  Parse x as $(c_1, \ldots, c_n)$
  $K_2 \leftarrow \mathcal{D}_K^{\mathrm{kr}}(c_1)$
  For $i = 2$ to n
    $K_{i \bmod n+1} \leftarrow \mathsf{Dec}'(K_i, c_i)$
  Add K_1 to L

$\underline{\mathcal{D}^{\mathrm{ae}}_{K,b}(c)}$
  If $b = 0$ then
    Return $\perp$
  Else
    Parse c as $\tilde{K} \parallel \tilde{m}$
    Add $\tilde{K}$ to L
    Return $\mathcal{D}_K^{\mathrm{kr}}(c)$

When $\mathcal{A}$ halts, $\mathcal{B}$ selects and outputs $\hat{K}$ at random from L.

Before moving on, let us intuitively explain how $\mathcal{B}$ is simulating the game. We have implemented the oracle simulation so that $\mathcal{B}$ assumes that the "If" statements in both oracles do not ever pass, and indeed it properly simulates both hybrids as long as this is case. It keeps track of the keys induced by the queries of $\mathcal{A}$ which might have caused an "If" statement to pass, and afterwards it chooses a random one and hopes it was the first such query.

Let E be the event that $\mathcal{A}$ queries either $\mathcal{E}^{\mathsf{ae}}_{K,b}(\cdot,\cdot)$ or $\mathcal{D}^{\mathsf{ae}}_{K,b}(\cdot)$ at a point that causes their "If" statements to evaluate to true. It is apparent that H_0 and H_1 are identical unless E occurs, so we have

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) \leq \Pr[E].$$

Conditioned on E occurring, we have that $\hat{K}$ is equal to K (where K was chosen in the KR experiment) with probability $1/Q$, where Q is (polynomial) number of queries issued by $\mathcal{A}$. This follows from the fact that $\mathcal{B}$ perfectly simulates H_0 until the first query that triggers the event E. Thus, $\mathcal{B}$ recovers the secret key with probability at least $\mathsf{Adv}^{\mathsf{kr}}_{\Pi'_{\mathsf{ae}},\mathcal{B}}(\lambda) = \Pr[E]/Q$. But this is negligible by the assumption that Π'_{ae} is AE-secure and hence KR-secure, which bounds $\Pr[E]$ by a negligible function. □

Lemma A.5 *For every PPT adversary* $\mathcal{A}$

$$\mathsf{Adv}^{\mathsf{ae}}_{\Pi'_{\mathsf{ae}},\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda). \tag{3}$$

Proof. This lemma follows by the observation that in H_1, $\mathcal{A}$ is interacting with oracles that are functionally identical to those in $\mathsf{AE}(\Pi'_{\mathsf{ae}},\mathcal{A},\lambda)$. The only difference is in the message space restriction in H_1, which is a strict subset of those allowed in $\mathsf{AE}(\Pi'_{\mathsf{ae}},\mathcal{A},\lambda)$. □

Finally, we observe that $\mathsf{Adv}^{\mathsf{ae}}_{\Pi'_{\mathsf{ae}},\mathcal{B}}(\lambda)$ is negligible by assumption. Combining this observation with (1), (2) and (3) proves the theorem. □

B Security Proof for System Π_{cpa}

We first recall Theorem 4.3.

Theorem B.1 *Encryption scheme* Π_{cpa} *is* $\mathsf{IND\text{-}CPA}$ *secure under the Decisional Diffie-Hellman Assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *(SXDH).*

Proof. To show that scheme Π_{cpa} meets security Definition 2.1, suppose PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has advantage ϵ in the $\mathsf{IND\text{-}CPA}(\Pi_{\mathsf{cpa}},\mathcal{A},\lambda)$ experiment. Let $\psi(\cdot)$ be some polynomial function that will be determined in the proof. Using a series of hybrid games we show that if all PPT adversaries have negligible advantage ϵ_1 in solving the DDH problem in $\mathbb{G}_1$ or $\mathbb{G}_2$ and advantage $\psi(\epsilon_1)$ at distinguishing the PRG F (secure under DDH) from a random function, then ϵ is bounded by the negligible value $4\epsilon_1 + 2\psi(\epsilon_1)$.

In all hybrids, the adversary plays the $\mathsf{IND\text{-}CPA}$ game with a challenger. The public key is distributed normally, but the structure of the challenge ciphertext differs between the hybrids. Let $\mathsf{CT} = (C_1, C_2, C_3, C_4)$ denote the challenge ciphertext computed in $\mathsf{IND\text{-}CPA}$ and let $R_2 \xleftarrow{r} \mathbb{G}_T$, $R_3 \xleftarrow{r} \mathbb{G}_1$ (if $\beta = 0$) or $R_3 \xleftarrow{r} \mathbb{G}_2$ (if $\beta = 1$) and $R_4 \xleftarrow{r} \{0,1\}^{|C_4|}$ be randomly chosen. The hybrids are as follows:

H_0: The challenge ciphertext is $\mathsf{CT} = (C_1, C_2, C_3, C_4)$.
H_1: The challenge ciphertext is $\mathsf{CT}_1 = (C_1, R_2, C_3, C_4)$.
H_2: The challenge ciphertext is $\mathsf{CT}_2 = (C_1, R_2, R_3, C_4)$.
H_3: The challenge ciphertext is $\mathsf{CT}_3 = (C_1, R_2, R_3, R_4)$.

We will write $\mathsf{Adv}^{\mathsf{H}_i}_{\mathcal{A}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in H_i, i.e., $2\Pr[\mathsf{H}_i(\mathcal{A},\lambda) = 1] - 1$. By definition, the ciphertext in H_0 is as in $\mathsf{IND\text{-}CPA}(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$, while the challenge ciphertext in hybrid H_3 information-theoretically hides the plaintext. We argue that under the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, for all PPT $\mathcal{A}$, we have

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}} - \mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}} \leq 2\epsilon_1 + \psi(\epsilon_1). \tag{4}$$

It remains to observe that, by definition,

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) = \mathsf{IND\text{-}CPA}(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda), \tag{5}$$

and

$$\mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}}(\lambda) = 0 \tag{6}$$

because the adversary's output is independent of the bit b it is trying to guess.

We now turn to proving (4). We start by bounding the difference in advantage between H_0 and H_1.

Lemma B.2 *For all PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then*

$$\mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) \leq \epsilon_1.$$

Proof. Let $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g = \langle \mathbb{G}_1 \rangle, h = \langle \mathbb{G}_2 \rangle)$ be the common parameters. Suppose for contradiction that an adversary $\mathcal{A}$ violates the inequality in the lemma. Then, we construct an adversary $\mathcal{A}'$ that decides the DDH problem in $\mathbb{G}_1$ or $\mathbb{G}_2$ with advantage ϵ'. $\mathcal{A}'$ works as follows.

1. Sample a bit $\beta \leftarrow \{0,1\}$.
2. Obtain a DDH problem instance:
$$\Gamma = \begin{cases} (g, g^a, g^b, G) \in \mathbb{G}_1^4 & \text{if } \beta = 0; \\ (h, h^a, h^b, H) \in \mathbb{G}_2^4 & \text{if } \beta = 1. \end{cases}$$
3. Sample $v \leftarrow \mathbb{Z}_p^*$.
4. Set the public key as:
$$pk = \begin{cases} (0, e(g^a, h), g^v) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_1 & \text{if } \beta = 0; \\ (1, e(g, h^a), h^v) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_2 & \text{if } \beta = 1. \end{cases}$$
5. Run $\mathcal{A}_1(pk)$ to produce a tuple (M_0, M_1, z). Parse M_0 as (α, m_1, m_2).
6. Sample $R \leftarrow \mathbb{G}_T$ and set $I \leftarrow F(R) \oplus \mathsf{encode}(M_0)$.
7. Set the challenge ciphertext as:
$$C = \begin{cases} (g^b,\ R \cdot e(G,h),\ (g^b)^{vm_2} \cdot g^{m_1},\ I) & \text{if } \beta = 0; \\ (h^b,\ R \cdot e(g,H),\ (h^b)^{vm_2}\ \ \ \ \ \ ,\ I) & \text{if } \beta = 1. \end{cases}$$
Note that in the first case, $C \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)}$, while in the second case $C \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)}$.
8. Run $\mathcal{A}_2(C, z)$ and output whatever it outputs.

We argue that when Γ is a proper DDH instance, $\mathcal{A}'$ perfectly simulates the experiment H_0. The distribution of keys and encryption values are exactly as they should be. When Γ is not a DDH instance, $\mathcal{A}'$ perfectly simulates the experiment H_1. The only impacted ciphertext part is C_2, where the proper public key information has been replaced by a random value. Thus, $\mathcal{A}'$'s advantage in solving DDH in $\mathbb{G}_1$ or $\mathbb{G}_2$ will be ϵ'. Under the DDH assumption in $\mathbb{G}_1, \mathbb{G}_2$, $\epsilon' \leq \epsilon_1$. □

Lemma B.3 *For all PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then*

$$\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) \leq \epsilon_1.$$

Proof. Suppose adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ violates the lemma. Then, we construct an adversary $\mathcal{A}'$ that decides the DDH problem in $\mathbb{G}_1$ or $\mathbb{G}_2$ with advantage ϵ' as follows. Let $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g = \langle \mathbb{G}_1 \rangle, h = \langle \mathbb{G}_2 \rangle)$ be the common parameters. $\mathcal{A}'$ works as follows:

1. Sample a bit $\beta \leftarrow \{0, 1\}$.
2. Obtain a DDH problem instance:

$$\Gamma = \begin{cases} (g, g^a, g^b, G) \in \mathbb{G}_1^4 & \text{if } \beta = 0; \\ (h, h^a, h^b, H) \in \mathbb{G}_2^4 & \text{if } \beta = 1. \end{cases}$$

3. Sample $v \leftarrow \mathbb{Z}_p^*$.
4. Set the public key as:

$$pk = \begin{cases} (0, e(g,h)^v, g^a) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_1 & \text{if } \beta = 0; \\ (1, e(g,h)^v, h^a) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_2 & \text{if } \beta = 1. \end{cases}$$

5. Run $\mathcal{A}_1(pk)$ to produce a tuple (M_0, M_1, z). Parse M_0 as (α, m_1, m_2).
6. Sample $R, R_2 \leftarrow \mathbb{G}_T$ and set $I \leftarrow F(R) \oplus \mathsf{encode}(M_0)$.
7. Set the challenge ciphertext as:

$$C = \begin{cases} (g^b,\ R_2,\ G^{m_2} \cdot g^{m_1},\ I) & \text{if } \beta = 0; \\ (h^b,\ R_2,\ H^{m_2}\ \ \ \ ,\ I) & \text{if } \beta = 1. \end{cases}$$

8. Run $\mathcal{A}_2(C, z)$ and output whatever it outputs.

When Γ is a proper DDH instance, $\mathcal{A}'$ perfectly simulates experiment H_1. When Γ is not a DDH instance, $\mathcal{A}'$ perfectly simulates experiment H_2. The only impacted ciphertext part is C_3, where the proper public key information has been replaced by a random value. Thus, $\mathcal{A}'$'s advantage in solving DDH in $\mathbb{G}_1$ or $\mathbb{G}_2$ will be ϵ'. Under the DDH assumption in $\mathbb{G}_1, \mathbb{G}_2$, $\epsilon' \leq \epsilon_1$. □

Lemma B.4 *For all PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ if F is secure under the DDH assumption in $\mathbb{G}_1, \mathbb{G}_2$ then*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_2}(\lambda) \leq \psi(\epsilon_1).$$

Proof. Let $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g = \langle \mathbb{G}_1 \rangle, h = \langle \mathbb{G}_2 \rangle)$ be the common parameters. Note that in our construction, F has domain $\mathbb{G}_T$ and range $\{0,1\}^{\ell(\lambda)}$.[4] Let us suppose that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ violates the lemma. Then, we construct an adversary $\mathcal{A}'$ that breaks the security of the PRG F with advantage ϵ'. $\mathcal{A}'$ accepts as input a value I' sampled from ensemble E_b where $E_0 = \{R \leftarrow \mathbb{G}_T : F(R)\}_\lambda$, $E_1 = \{U_{\ell(\lambda)}\}_\lambda$ and $b \in \{0,1\}$ and operates as follows:

1. Compute $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$ and parse $pk = (\beta, Y_1, Y_2)$.
2. Run $\mathcal{A}_1(pk)$ to produce a tuple (M_0, M_1, z).
3. Sample $r \leftarrow \mathbb{Z}_p$, $R_2 \leftarrow \mathbb{G}_T$ and $R_3 \leftarrow \mathbb{G}_1$ (if $\beta = 0$) or $R_3 \leftarrow \mathbb{G}_2$ (if $\beta = 1$). Set $I \leftarrow I' \oplus \mathsf{encode}(M_0)$. Compute the challenge ciphertext as follows:

$$C = \begin{cases} (g^r,\ R_2,\ R_3,\ I) & \text{if } \beta = 0; \\ (h^r,\ R_2,\ R_3,\ I) & \text{if } \beta = 1. \end{cases}$$

4. Run $\mathcal{A}_2(C, z)$ and output whatever it outputs.

[4] Although this specification differs slightly from Definition 4.2, this specific construction can be constructed from traditional PRGs using standard techniques.

If I' is sampled from distribution E_0 then $\mathcal{A}'$ perfectly simulates H_2. If I' is sampled from the uniform distribution E_1, then $I' \oplus \mathsf{encode}(M_0)$ is uniformly distributed in $\{0,1\}^{\ell(\lambda)}$ and $\mathcal{A}'$ perfectly simulates H_3. Additionally, R is independent of the adversary's view. Thus $\mathcal{A}'$'s advantage in distinguishing the two distributions will be ϵ'. Under the DDH assumption, we have $\epsilon' \leq \psi(\epsilon_1)$. □

We complete the proof of the theorem by combining (4), (5), (6), and Lemmas B.2, B.3, and B.4. □

C An Alternative Counterexample for CPA Security

As mentioned in Section 4, one "artificial" feature of the cryptosystem Π_{cpa} is that the KeyGen algorithm randomly embeds the public key into either $\mathbb{G}_1$ or $\mathbb{G}_2$ with probability 1/2 and then the group setting of the ciphertext also differs depending on the public key. We know of no deployed cryptosystems that alternate the setting of keys in such a manner. Some readers might hope that this property renders our result inapplicable to the domain of "practical" cryptosystems, i.e., to assume that cryptosystems with a single, defined key and ciphertext structure are immune to the concerns we note here. We must disappoint these readers.

Below we propose an alternative IND-CPA-secure scheme Π'_{cpa} that does not exhibit this "group switching" feature, and yet still breaks catastrophically in the face of a 2-cycle. *Indeed, this result is even stronger than that of Section 4 since it permits an adversary to win the* IND-CIRC-CPA *game with a higher probability.* Π'_{cpa} has keys and ciphertexts that are twice the length of those in Π_{cpa}.

Construction Π'_{cpa} Cryptosystem $\Pi'_{\mathsf{cpa}} = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ uses $\Pi_{\mathsf{cpa}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as a building block. As before we assume that a single set of bilinear group parameters will be shared across all keys generated at a given security level and are implicitly provided to all algorithms. Let $\mathcal{M}$ be the message space of Π_{cpa}. Then the message space for Π'_{cpa} is $\mathcal{M}' = \mathcal{M} \times \mathcal{M}$. We define the system as follows.

$\mathsf{KeyGen}'(1^\lambda)$. The key generation algorithm runs KeyGen repeatedly to obtain pk_1, sk_1 and pk_2, sk_2 where $pk_1 = (0, \cdot, \cdot)$ and $pk_2 = (1, \cdot, \cdot)$.[5] The public key is set as $pk = (pk_1, pk_2)$, and the secret key as $sk = (sk_1, sk_2)$.

$\mathsf{Encrypt}'(pk, M)$. The encryption algorithm parses the public key $pk = (pk_1, pk_2)$, and message $M = (m_1, m_2) \in \mathcal{M}'$. Output the ciphertext C as:

$$C = (\mathsf{Enc}(pk_1, m_2), \mathsf{Enc}(pk_2, m_1))$$

$\mathsf{Decrypt}'(sk, C)$. The decryption algorithm parses the secret key $sk = (sk_1, sk_2)$ and the ciphertext $C = (C_1, C_2)$. Next, it computes:

$$M = (\mathsf{Dec}(sk_2, C_2), \mathsf{Dec}(sk_1, C_1))$$

Correctness follows trivially from the correctness of Π_{cpa}.

Theorem C.1 *Encryption scheme* Π'_{cpa} *is* IND-CPA *secure under the Decisional Diffie-Hellman Assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *(SXDH).*

Attack on IND-CIRC-CPA **Security** The above scheme breaks completely for 2-key cycles.

Theorem C.2 *Encryption scheme* Π'_{cpa} *is not* IND-CIRC-CPA *secure for cycles of length 2.*

Proof sketch. To show that scheme Π'_{cpa} is *not* IND-CIRC-CPA-secure for key cycles of length two, we recall the attack of Section 4.3. As in that attack, we assume that the adversary receives $C_A = \mathsf{Enc}(pk_A, sk_B)$ and $C_B = \mathsf{Enc}(pk_B, sk_A)$ or two encryptions of a fixed message, and must distinguish which. Unlike that attack, we do not abort based on the structure of the public keys. Instead we receive $pk_A = (pk_{A,1}, pk_{A,2})$, $pk_B = (pk_{B,1}, pk_{B,2})$, $C_A = (C_{A,1}, C_{A,2})$ and $C_B = (C_{B,1}, C_{B,2})$. Now, there are two options. Either:

[5] This can be accomplished probabilistically by repeatedly calling KeyGen and discarding redundant keypairs; alternatively the KeyGen algorithm can be trivially modified to produce the needed keys in only two calls.

1. $C_{A,1} = \mathsf{Enc}(pk_{A,1}, sk_{B,2})$ and $C_{B,2} = \mathsf{Enc}(pk_{B,2}, sk_{A,1})$ and
   $C_{A,2} = \mathsf{Enc}(pk_{A,2}, sk_{B,1})$ and $C_{B,1} = \mathsf{Enc}(pk_{B,1}, sk_{A,2})$; or

2. $C_{A,1} = \mathsf{Enc}(pk_{A,1}, \alpha_2)$ and $C_{B,2} = \mathsf{Enc}(pk_{B,2}, \alpha_1)$ and
   $C_{A,2} = \mathsf{Enc}(pk_{A,2}, \alpha_1)$ and $C_{B,1} = \mathsf{Enc}(pk_{B,1}, \alpha_2)$
   for any fixed $(\alpha_1, \alpha_2) \in \mathcal{M}'$ as defined by Definition 2.4.

If we are in case 1, then we simply apply the exact attack from Section 4.3 twice to the pairs $(C_{A,1}, C_{B,2})$ and $(C_{A,2}, C_{B,1})$ to recover both secret keys in full $(sk_{A,1}, sk_{A,2})$ and $(sk_{B,1}, sk_{B,2})$ with probability 1. Once this is done and detected, D outputs 1.

If we are in case 2, then let $\alpha_1 = (\cdot, m_1, m_2)$ and $\alpha_2 = (\cdot, m_1', m_2')$. Parse $sk_{A,1} = (0, a_1, a_2)$ and $sk_{B,2} = (1, b_1, b_2)$ and we have:

$$\begin{aligned}
C_{A,1} =& (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4}) \\
=& (g^r, R \cdot e(g,h)^{ra_1}, g^{ra_2 m_2' + m_1'}, F(R) \oplus \mathsf{encode}(\alpha_2)) \\
C_{B,2} =& (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4}) \\
=& (h^s, S \cdot e(g,h)^{sb_1}, h^{sm_2 b_2}, F(S) \oplus \mathsf{encode}(\alpha_1))
\end{aligned}$$

for some $r, s \in \mathbb{Z}_p$ and $R, S \in \mathbb{G}_T$. Then we have that:

$$\begin{aligned}
X &:= c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})} = S \cdot e(g,h)^{sb_1} \cdot \frac{e(g^r, h^{sm_2 b_2})}{e(g^{ra_2 m_2' + m_1'}, h^s)} \\
&= S \cdot e(g,h)^{s(b_1 - m_1')} \cdot (e(g,h)^{s(m_2 b_2 - m_2' a_2)})^r
\end{aligned}$$

Now, D will return 1 if and only if $sk_A = \mathsf{decode}((F(S) \oplus \mathsf{encode}(\alpha_1)) \oplus F(X))$. What is the probability that this event occurs? First, suppose that $s(m_2 b_2 - m_2' a_2) \bmod p \neq 0$ (event E_1), which happens with probability $\geq 1 - 3/(p-1) = (p-4)/(p-1)$ for honest executions. Next, consider the values α_1, α_2, s, S as fixed and r is the only variable. What is the chance that the challenger's random choice of r will induce a value X such that $F(X) = F(S) \oplus \mathsf{encode}(\alpha_1) \oplus \mathsf{encode}(sk_A)$? First, we observe that since $s(m_2 b_2 - m_2' a_2) \neq 0$ and r is chosen uniformly at random in $\mathbb{Z}_p$, then X is also distributed uniformly at random in $\mathbb{G}_T$. Thus, by the assumption that F is computationally indistinguishable from a uniform, random function, D will incorrectly guess a key cycle in this case with probability at most $2^{-\ell(\lambda)}$ plus a negligible amount $\nu(\lambda)$, where λ is the security parameter.

Thus, D's total probability of success in this attack is:

$$\begin{aligned}
\Pr[D \text{ wins}] =& \Pr[\text{Case 1}] \cdot \Pr[D \text{ wins } | \text{Case 1}] \\
&+ \Pr[\text{Case 2}] \cdot \Pr[D \text{ wins } | \text{Case 2}] \\
\geq& \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (\Pr[E_1] \cdot \Pr[D \text{ wins } | E_1]) \\
\geq& \frac{1}{2} + \frac{1}{2} \cdot \left(\frac{p-4}{p-1} \cdot (1 - 2^{-\ell(\lambda)} - \nu(\lambda)) \right) \\
\geq& \frac{3}{4} - \frac{(2^{-\ell(\lambda)} + \nu(\lambda))}{2} \quad \text{for all } p \geq 7
\end{aligned}$$

Of course, for practical 80-bit or higher values of p, this probability is much closer to 1. □