CPA and CCA-Secure Encryption Systems that are not 2-Circular Secure

Matthew Green*
Johns Hopkins University

Susan Hohenberger*
Johns Hopkins University

Abstract

Traditional definitions of encryption guarantee security for plaintexts which can be derived by the adversary. In some settings, such as anonymous credential or disk encryption systems, one may need to reason about the security of messages potentially unknown to the adversary, such as secret keys encrypted in a self-loop or a cycle. A public-key cryptosystem is *n-circular secure* if it remains secure when the ciphertexts $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ are revealed, for independent key pairs.

A natural question to ask is what does it take to realize circular security in the standard model? Are all CPA-secure (or CCA-secure) cryptosystems also n-circular secure for $n > 1$? One way to resolve this question is to produce a CPA-secure (or CCA-secure) cryptosystem which is demonstrably insecure for key cycles larger than self-loops. Recently and independently, Acar, Belenkiy, Bellare and Cash provided a CPA-secure cryptosystem, under the SXDH assumption, that is not 2-circular secure.

In this paper, we present a different CPA-secure counterexample (under SXDH) as well as the first CCA-secure counterexample (under SXDH and the existence of certain NIZK proof systems) for $n > 1$. Moreover, our 2-circular attacks recover the secret keys of both parties and thus exhibit a catastrophic failure of the system whereas the attack in Acar et al. provides a test whereby the adversary can distinguish whether it is given a 2-cycle or two random ciphertexts. These negative results are an important step in answering deep questions about which attacks are prevented by commonly-used definitions and systems of encryption.

1 Introduction

Encryption is one of the most fundamental cryptographic primitives. Traditional definitions of encryption [22, 17, 35] follow the seminal notion of Goldwasser and Micali which guarantees indistinguishability of encryptions for messages chosen by the adversary [22]. 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., [13, 9, 31, 3, 27, 29, 11, 12, 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 [13]. A public-key cryptosystem is *n-circular secure* if it remains secure when the ciphertexts

*This work was supported by NSF CNS-0716142, Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641) and a Microsoft Research New Faculty Fellowship.

$E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ are revealed, for independent key pairs. Either by design or accident, these key cycles naturally arise in many applications, including storage systems such as BitLocker [11], anonymous credentials [13], the study of "axiomatic security" [31, 3] and more. (See [11] for a fuller 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. While the recent work of Haitner and Holenstein casts some doubt on our ability to prove this implication using standard black-box techniques [26], it does not resolve the fundamental question of whether or not this implication is, in fact, true.

Recently, this gap has been closing in two ways. On the positive side, several circular-secure encryption schemes have been proposed [11, 5, 12]. This work focuses on negative results – namely, investigating whether standard notions of encryption are "safe" for circular applications.

In 2008, Boneh, Halevi, Hamburg and Ostrovsky [11] proved, by counterexample, that *one-way* security does not imply circular security. They explicitly state that they would ideally like to have a counterexample for CPA security, but were not able to find a candidate system. The CPA setting is significantly more difficult than the one-way setting, because all parts of the message must be "hidden" in the ciphertext, so there appears to be no natural extension of the above trick. *One must find a method for combining (truly secure) ciphertexts, generated independently with unique randomness, to recover information about their underlying messages.* At first glance, this seems like it might be impossible.

Our Results. A fundamental question for encryption is: do today's widely-used definitions imply that it is safe for Alice and Bob to exchange the ciphertexts $E(pk_B, sk_A)$ and $E(pk_A, sk_B)$ over an insecure channel? In this work, we answer *no*, for both the CPA and CCA-security notions, by providing counterexamples relative to popular cryptographic assumptions. Specifically, we show that:

- If there exists an algebraic setting where the Symmetric External Diffie-Hellman (SXDH) assumption holds, then there exists a CPA-secure cryptosystem which is *not* 2-circular secure. The proposed scheme is particular interesting in that it breaks *catastrophically* in the presence of a 2-cycle — revealing the secret keys of both users. This is a significant counterexample separating CPA and circular security for key cycles larger than self-loops (where such a counterexample is trivial.)

  In our El Gamal-inspired system, the adversary can distinguish encryptions of key cycles from encryptions of zero with probability at least 5/8 minus a negligible amount and, if given encryptions of key cycles, can recover both secret keys with probability 1/2. In Appendix A, we show how to extend our system to improve both probabilities to almost 1.

- 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-circular secure.

These results deepen our understanding of how to define "secure" encryption. Relative to SXDH, we rule out the possibility that CPA security inherently captures n-circular security for $n = 2$, and

thus provide strong justification for the ongoing effort, e.g. [11, 12, 5], to develop cryptosystems which are provably circular secure. Our results are presented in the public-key setting, but trivially transfer to the symmetric setting by having both parties share a secret key.

The SXDH assumption states that there is a bilinear setting $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$. It has been extensively studied and used e.g., [19, 37, 32, 10, 7, 6, 23, 8, 25], perhaps most notably as a setting of the Groth-Sahai NIZK proof system [25].

Thus, we are faced with at least one of two harsh realities. Either the SXDH assumption is false (and many prior constructions may be broken) or our standard definition of encryption, CPA-security, does not guarantee n-circular security for $n = 2$, with fresh doubt cast on all $n > 2$.

Moreover, our two counterexample constructions are simple and practical, i.e., similar to what might conceivably be proposed in practice, which highlights the potential danger in using any of today's commonly-used cryptosystems in a circular setting without further analysis.

1.1 Related Work

In 2001, Camenisch and Lysyanskaya [13] 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 [9] 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 [27] 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 [29] showed how to construct circular-secure cryptosystems in the standard model under relaxed notions of security.

In the public-key setting, Boneh, Halevi, Hamburg and Ostrovsky [11] presented in 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. [11] 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 $\mathsf{Encrypt}(pk_A, sk_B) = (sk_{B,1}, E(pk_{A,1}, sk_{B,2}))$ and $\mathsf{Encrypt}(pk_B, sk_A) = (sk_{A,1}, E(pk_{B,1}, sk_{A,2}))$ provide the critical secret key information $(sk_{B,1}, sk_{A,1})$ in the clear.

Subsequently, Applebaum, Cash, Peikert and Sahai [5] showed how to translate the circular-secure construction of [11] into the lattice setting. In addition, Camenisch, Chandran and Shoup [12] extended [11] 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 [34]. (Interestingly, we use this same approach in Section 4 to extend our counterexample from CPA to CCA security.)

Haitner and Holenstein [26] recently provided strong impossibility results for KDM-security *with respect to 1-key cycles*. 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 reductions proof of security treats both the adversary and the function g as black boxes. Another way to compare these results to ours is to say that they provide negative results for "self-loop" KDM-security relative to the non-existence of certain non-black-box reduction techniques, whereas we provide negative results for "two-party" circular security relative to falsifiable number-theoretic assumptions.

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 [33], 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 [16], 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 $E(pk, m)||0$ 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 [11].

CPA Counterexample of Acar, Belenkiy, Bellare and Cash. Recently and independently of our work, Acar et al. [2] also demonstrated both public and private key encryption systems that are provably CPA-secure and yet also demonstrably 2-circular *insecure*. Their constructions also depend on the SXDH assumption. Our works differ in two primary ways. First, Acar et al. break 2-circular security by providing an elegant distinguishing test for the adversary to differentiate the pair $(\mathsf{Encrypt}(pk_A, sk_B), \mathsf{Encrypt}(pk_B, sk_A))$ from the pair $(\mathsf{Encrypt}(pk_A, r_1), \mathsf{Encrypt}(pk_B, r_2))$, where (r_1, r_2) are random values in the message space. In this work, we provide a stronger attack, which not only allows the adversary to distinguish between these pairs, but also allows the adversary to *recover the secret keys* (sk_A, sk_B) when given $(\mathsf{Encrypt}(pk_A, sk_B), \mathsf{Encrypt}(pk_B, sk_A))$. In other words, Acar et al. show that CPA-security does not prevent an eavesdropper from detecting that Alice and Bob have exchanged secret keys via circular encryptions, whereas we show that CPA-security provides *no security guarantee at all* in the same setting. Second, we provide the first counterexample that also extends to CCA-security.

2 Definitions of Security

A public-key encryption system Π is a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, where KeyGen is a key-generation algorithm that takes as input a security parameter λ and outputs a public/secret key pair (pk, sk); $\mathsf{Encrypt}(pk, m)$ encrypts a message m under public key pk; and $\mathsf{Decrypt}(sk, c)$

decrypts ciphertext c with secret key sk. As in most other works, we assume that all algorithms implicitly have access to shared public parameters establishing a common algebraic setting.

Throughout this paper, we assume that the space of secret keys output by KeyGen is a subset of the message space and thus any secret key can be encrypted using any public key.

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

Definition 2.1 (Computational 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*

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

We denote such sets $\{X_k\}_{k\in\mathbb{N}} \stackrel{c}{\approx} \{Y_k\}_{k\in\mathbb{N}}$.

2.1 Standard Indistinguishability of Encryptions

We recall the standard notion of indistinguishability of encryptions under a chosen-plaintext attack due to Goldwasser and Micali [22].

Definition 2.2 ($\mathsf{IND\text{-}CPA}$) *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an encryption scheme for the message space M and let the random variable $\mathsf{IND\text{-}CPA}_b(\Pi, \mathcal{A}, \lambda)$ where $b \in \{0,1\}$, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\lambda \in \mathbb{N}$ denote the result of the following probabilistic experiment:*

$\mathsf{IND\text{-}CCA}_b(\Pi, \mathcal{A}, \lambda)$
  $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  $(m_0, m_1, z) \leftarrow \mathcal{A}_1(pk)$ *s.t.* $m_0, m_1 \in M$
  $y \leftarrow \mathsf{Encrypt}(pk, m_b)$
  $B \leftarrow \mathcal{A}_2(y, z)$
  Output B

Encryption scheme Π is $\mathsf{IND\text{-}CPA}$-secure if $\forall$ p.p.t. algorithms $\mathcal{A}$ the following two ensembles are computationally indistinguishable:

$$\left\{\mathsf{IND\text{-}CPA}_0(\Pi, \mathcal{A}, \lambda)\right\}_\lambda \stackrel{c}{\approx} \left\{\mathsf{IND\text{-}CPA}_1(\Pi, \mathcal{A}, \lambda)\right\}_\lambda$$

We also consider the indistinguishability of encryptions under chosen-ciphertext attacks [34, 35, 17].

Definition 2.3 ($\mathsf{IND\text{-}CCA}$) *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an encryption scheme for the message space M and let the random variable $\mathsf{IND\text{-}CCA}_b(\Pi, \mathcal{A}, \lambda)$ be identical to $\mathsf{IND\text{-}CPA}_b(\Pi, \mathcal{A}, \lambda)$ except that both $\mathcal{A}_1$ and $\mathcal{A}_2$ have access to an oracle $\mathsf{Decrypt}(sk, \cdot)$ that returns the output of the decryption algorithm and $\mathcal{A}_2$ cannot query this oracle on input y.*

Encryption scheme Π is $\mathsf{IND\text{-}CCA}$-secure if $\forall$ p.p.t. algorithms $\mathcal{A}$ the following two ensembles are computationally indistinguishable:

$$\left\{\mathsf{IND\text{-}CCA}_0(\Pi, \mathcal{A}, \lambda)\right\}_\lambda \stackrel{c}{\approx} \left\{\mathsf{IND\text{-}CCA}_1(\Pi, \mathcal{A}, \lambda)\right\}_\lambda$$

2.2 Circular Security

We next recall the Key-Dependent Message (KDM) security notion of Black et al. [9]. We simplify this definition to focus exclusively on key cycles, as opposed to any affine function of the secret keys as in [27, 11]. In [11], Boneh et al. were proving a positive result and thus wanted to demonstrate the robustness of their construction by giving the adversary as much power as possible. In this work, we are proving a negative result. By restricting the adversary's power, we make it significantly harder for us to devise a counterexample and thus prove a much stronger result.[1]

Definition 2.4 (IND-CIRC-CPA) *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *be an encryption scheme for the message space* M *and let the random variable* $\mathsf{IND\text{-}CIRC\text{-}CPA}_b^n(\Pi, \mathcal{A}, \lambda)$ *where* $b \in \{0,1\}$*, integer* $n > 0$*,* $\mathcal{A}$ *and* $\lambda \in \mathbb{N}$ *denote the result of the following probabilistic experiment:*

$\mathsf{IND\text{-}CIRC\text{-}CPA}_b^n(\Pi, \mathcal{A}, \lambda)$
  $(pk_1, sk_1) \leftarrow \mathsf{KeyGen}(1^\lambda), \ldots, (pk_n, sk_n) \leftarrow \mathsf{KeyGen}(1^\lambda)$
  For $i = 1$ *to* n*:*
    $m_{1,i} \leftarrow sk_{(i+1) \mod n}$
    $m_{0,i} \leftarrow 0^{|m_{1,i}|}$
    $y_i \leftarrow \mathsf{Encrypt}(pk_i, m_{b,i})$
  $B \leftarrow \mathcal{A}(pk_1, \ldots, pk_n, y_1, \ldots, y_n)$
  Output B

Encryption scheme Π *is* IND-CIRC-CPA*-secure for cycles of length* n *if* $\forall$ *p.p.t. algorithms* $\mathcal{A}$ *the following two ensembles are computationally indistinguishable:*

$$\left\{\mathsf{IND\text{-}CIRC\text{-}CPA}_0^n(\Pi, \mathcal{A}, \lambda)\right\}_\lambda \overset{c}{\approx} \left\{\mathsf{IND\text{-}CIRC\text{-}CPA}_1^n(\Pi, \mathcal{A}, \lambda)\right\}_\lambda$$

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 [11]; 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 the encryption of zero above is replaced by the encryption of 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, we provide an IND-CCA-secure cryptosystem, which is provably not IND-CIRC-CPA-secure. In other words, we are able to devise an peculiar cryptosystem: one that withstands all chosen-ciphertext attacks, and yet breaks under a weak circular attack which does not require a decryption oracle.

2.3 Pseudorandom Generators

Our constructions of Section 3 make use of a pseudorandom generator (PRG), which can be constructed from any one-way function [28].

[1] If we allowed the adversary to obtain encryptions of any affine function of the secret keys, as is done in [27, 11], then we could devise a trivial counterexample where the adversary uses 1-cycles to break the system.

Definition 2.5 (Pseudorandom Generator [30]) *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, $\{U_{\ell(n)}\}_n \stackrel{c}{\approx} \{s \leftarrow U_n : G(s)\}_n$.*

Note that the constructions of Section 3 use a PRG that differs slightly from this definition, in that the domain of the function is an exponentially-sized cyclic group.

3 Main Result: A Counterexample for CPA Security

3.1 Algebraic Setting

Bilinear Groups. We work in a bilinear setting where there exists an efficient mapping function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 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 p.p.t. adversaries $\mathcal{A}$, the following probability is $1/2$ plus an amount negligible in λ:

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

Strong External Diffie-Hellman Assumption (SXDH): Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be bilinear groups. The SXDH assumption states that the Decisional Diffie-Hellman (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 has been studied and used in many prior works, e.g., [19, 37, 32, 10, 7, 6, 23, 8, 25]. It is one of the three settings of the Groth-Sahai NIZK proof system [25] and, as noted by Ghadafi, Smart and Warinschi [20], SXDH is the only algebraic setting considered practical.

3.2 Encryption Scheme Π_{cpa}

We now describe an encryption scheme $\Pi_{\mathsf{cpa}} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$. It is set in asymmetric bilinear groups $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 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 $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g = \langle \mathbb{G}_1 \rangle, h = \langle \mathbb{G}_2 \rangle)$ 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 $\mathsf{encode} : \mathcal{M} \rightarrow \{0,1\}^{\ell(\lambda)}$ and $\mathsf{decode} : \{0,1\}^{\ell(\lambda)} \rightarrow \mathcal{M}$ denote an invertible encoding scheme where $\ell(\lambda)$ is the polynomial length of the encoded message. Let $F : \mathbb{G}_T \rightarrow \{0,1\}^{\ell(\lambda)}$ be a pseudorandom generator secure under the Decisional Diffie Hellman assumption. (Recall that pseudorandom generators can be constructed from any one-way function [28].)

$\mathsf{KeyGen}(1^\lambda)$. 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:

$$C = \begin{cases} (g^r,\ R \cdot Y_1^r,\ Y_2^{rm_2} \cdot g^{m_1},\ I) \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 0; \\ (h^r,\ R \cdot Y_1^r,\ Y_2^{rm_2},\ I) \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 1. \end{cases}$$

$\mathsf{Decrypt}(sk, C)$. 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. [11], the above cryptosystem is a variation on El Gamal [18]. 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., [14, 21]), we can address these points as well.[2] In Appendix A, 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., [10]) 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 might also lead to a more natural counterexample, perhaps involving one of today's commonly-used encryption algorithms.

3.3 Security

Theorem 3.1 *Encryption scheme* Π_{cpa} *is* IND-CPA *secure under the Decisional Diffie-Hellman Assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *(SXDH).*

[2] As a side note, 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.

Proof. To show that scheme Π_{cpa} meets security Definition 2.2, suppose p.p.t. adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and distinguisher D have advantage ϵ in distinguishing $\mathsf{IND\text{-}CPA}_0(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$ from $\mathsf{IND\text{-}CPA}_1(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$. Let $\psi(\cdot)$ be some polynomial function. Using a series of hybrid games we show that if all p.p.t. 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 for $\mathsf{IND\text{-}CPA}_0$, let $\mathsf{CT}' = (C'_1, C'_2, C'_3, C'_4)$ denote the challenge ciphertext for $\mathsf{IND\text{-}CPA}_1$, and let $R_2 \leftarrow \mathbb{G}_T$, $R_3 \leftarrow \mathbb{G}_1$ (if $\beta = 0$) or $R_3 \leftarrow \mathbb{G}_2$ (if $\beta = 1$) and $R_4 \leftarrow \{0,1\}^{|C_4|}$ 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 $\mathsf{CT}_3 = (C_1, R_2, R_3, R_4)$.
H_4: The challenge ciphertext is $\mathsf{CT}_4 = (C'_1, R_2, R_3, R_4)$.
H_5: The challenge ciphertext is $\mathsf{CT}_5 = (C'_1, R_2, R_3, C'_4)$.
H_6: The challenge ciphertext is $\mathsf{CT}_6 = (C'_1, R_2, C'_3, C'_4)$.
H_7: The challenge ciphertext is $\mathsf{CT}' = (C'_1, C'_2, C'_3, C'_4)$.

Note that the ciphertext in H_0 is as in $\mathsf{IND\text{-}CPA}_0(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$, 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 $\mathbb{G}_2$ all p.p.t. $\mathcal{A}, D$ distinguish hybrids H_0 and H_3 with advantage $\leq 2\epsilon_1 + \psi(\epsilon_1)$. Using a symmetric argument we show the same holds for hybrids H_4 and H_7, the latter of which is identical to $\mathsf{IND\text{-}CPA}_1(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$. It remains only to observe that hybrids H_3 and H_4 are identically-distributed. By summation we obtain $\epsilon \leq 4\epsilon_1 + 2\psi(\epsilon_1)$.

Our proof proceeds via a series of lemmas. We define the notation $\mathsf{Adv}_{\mathsf{H}_i}(\mathcal{A}, D)$ to be $\mathcal{A}$'s and D's advantage in distinguishing hybrid H_i from $\mathsf{IND\text{-}CPA}_0(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$. Clearly $\mathsf{Adv}_{\mathsf{H}_0}(\mathcal{A}, D) = 0$.

Lemma 3.2 *For all p.p.t. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ then* $\mathsf{Adv}_{\mathsf{H}_1}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_0}(\mathcal{A}, D) \leq \epsilon_1$.

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 adversary $\mathcal{A}$ and distinguisher D have advantage ϵ' in distinguishing H_0 from H_1. 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.

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:

$$C = \begin{cases} (g^b,\ R \cdot e(G,h),\ (g^b)^{vm_2} \cdot g^{m_1},\ I) \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 0; \\ (h^b,\ R \cdot e(g,H),\ (h^b)^{vm_2} \quad\quad ,\ I) \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 1. \end{cases}$$

8. Run $\mathcal{A}_2(C, z)$ to obtain the output B.
9. Run $t \leftarrow D(B)$ and output t.

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 3.3 *For all p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ *if the DDH assumption holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *then* $\mathsf{Adv}_{\mathsf{H}_2}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_1}(\mathcal{A}, D) \leq \epsilon_1$.

Proof. Suppose adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and distinguisher D have advantage ϵ' in distinguishing H_1 from H_2. 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:

$$C = \begin{cases} (g^b,\ R_2,\ G^{m_2} \cdot g^{m_1},\ I) \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 0; \\ (h^b,\ R_2,\ H^{m_2} \quad\quad ,\ I) \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 1. \end{cases}$$

8. Run $\mathcal{A}_2(C, z)$ to obtain the output B.
9. Run $t \leftarrow D(B)$ and output t.

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 3.4 *For all p.p.t. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ if F is secure under the DDH assumption in $\mathbb{G}_1, \mathbb{G}_2$ then $\mathsf{Adv}_{\mathsf{H}_3}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_2}(\mathcal{A}, D) \leq \psi(\epsilon_1)$.*

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 $\{0,1\}^{\ell(\lambda)}$.[3] Let us suppose that adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and distinguisher D have advantage ϵ' in distinguishing H_2 from H_3. 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:

$$C = \begin{cases} (g^r,\ R_2,\ R_3,\ I) \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 0; \\ (h^r,\ R_2,\ R_3,\ I) \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)} & \text{if } \beta = 1. \end{cases}$$

4. Run $\mathcal{A}_2(C, z)$ to obtain the output B.
5. Run $t \leftarrow D(B)$ and output t.

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 $\epsilon' \leq \psi(\epsilon_1)$. □

Lemma 3.5 *For all $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and distinguishers D, $\mathsf{Adv}_{\mathsf{H}_4}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_3}(\mathcal{A}, D) = 0$.*

Proof. Both C_1 and C_1' are distributed uniformly at random in $\mathbb{G}_1$ or $\mathbb{G}_2$, depending on β, and independent from all other parts of the ciphertext in both hybrids. □

Proofs of below lemmas are identical to those of Lemmas 3.4, 3.3 and 3.2 (respectively) with the sole modification that message M_1 is used to formulate the challenge ciphertext rather than M_0.

Lemma 3.6 *For all p.p.t. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ if F is secure under the DDH problem in $\mathbb{G}_1, \mathbb{G}_2$ then $\mathsf{Adv}_{\mathsf{H}_5}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_4}(\mathcal{A}, D) \leq \psi(\epsilon_1)$.*

Lemma 3.7 *For all p.p.t. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ then $\mathsf{Adv}_{\mathsf{H}_6}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_5}(\mathcal{A}, D) \leq \epsilon_1$.*

[3] Although this specification differs slightly from Definition 2.5, this specific construction can be constructed from traditional PRGs using standard techniques.

Lemma 3.8 *For all p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), D$ *if the DDH assumption holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *then* $\mathsf{Adv}_{\mathsf{H}_7}(\mathcal{A}, D) - \mathsf{Adv}_{\mathsf{H}_6}(\mathcal{A}, D) \leq \epsilon_1$.

Thus, if p.p.t. adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and distinguisher D have advantage ϵ in distinguishing $\mathsf{IND\text{-}CPA}_0(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$ from $\mathsf{IND\text{-}CPA}_1(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)$ then by summing over the above hybrids, we obtain that $\epsilon \leq 4\epsilon_1 + 2\psi(\epsilon_1)$ for negligible ϵ_1 and $\psi(\epsilon_1)$. This concludes our proof. □

3.4 The Attack

Despite being IND-CPA-secure, cryptosystem Π_{cpa} breaks catastrophically under a 2-cycle. Specifically, Eve can distinguish the ciphertexts $\mathsf{Encrypt}(pk_A, sk_B)$ and $\mathsf{Encrypt}(pk_B, sk_A)$ from encryptions of an arbitrary message with probability almost 5/8. Moreover, if the encryptions are of the secret keys, then Eve can recover both sk_A and sk_B with probability 1/2. This is the first circular attack that allows the adversary to recover the secret keys. (In Appendix A, we discuss how to improve these probabilities to 1 minus a negligible amount.) 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 somewhat amazing 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 3.9 *Encryption scheme* Π_{cpa} *is not* IND-CIRC-CPA *secure for cycles of length 2.*

Proof. We demonstrate a simple attack that permits a p.p.t adversary $\mathcal{A}$ and a distinguisher D to win the IND-CIRC-CPA game for cycles of length two with probability at least 5/8 minus a negligible amount. Suppose a challenger honestly samples from $\{\mathsf{IND\text{-}CIRC\text{-}CPA}^2_b(\Pi_{\mathsf{cpa}}, \mathcal{A}, \lambda)\}_\lambda$ with $b \leftarrow \{0,1\}$ chosen with probability exactly 1/2. Then the adversary and distinguisher proceed as follows. The adversary $\mathcal{A}$ first obtains pk_A and pk_B. If both keys have $\beta = 0$ or $\beta = 1$ (event E_1), the adversary aborts and tells D to output a random bit. Since the two keys are honestly and independently generated by the challenger, this event will occur with probability exactly 1/2. Otherwise we will assume w.l.o.g. 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.

The adversary is also given ciphertexts $C_A = (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4})$ and $C_B = (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4})$. The adversary now computes:

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

$\mathcal{A}$ computes $M = \mathsf{decode}(c_{B,4} \oplus F(X))$ and passes it to D. D verifies that $M = sk_A$ by testing that $\alpha = 0$, $e(g,h)^{a_1} = e(g,h)^{m_1}$ and $g^{a_2} = g^{m_2}$. If all tests pass, D outputs 1, else it outputs 0.

Let's explore why this test works. First, suppose that $C_A = \mathsf{Encrypt}(pk_A, sk_B)$ and $C_B = \mathsf{Encrypt}(pk_B, sk_A)$ (event E_2). Then:

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

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

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

Thus, $\mathcal{A}$ can recover sk_A as $\mathsf{decode}(c_{B,4} \oplus F(S))$, and D will correctly answer 1 with probability 1.

Next, suppose that C_A and C_B encrypt an arbitrary constant $J = (\alpha, m_1, m_2) \in \mathcal{M}$. (In [27, 11] and Definitions 2.4, this constant is typically set to zero, but since zero is not in the message space of our cryptosystem, any other constant, such as all ones, will be equivalent.) Then:

$$\begin{aligned} C_A = \quad & (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4}) = \quad (g^r, R \cdot e(g,h)^{ra_1}, g^{ra_2m_2+m_1}, F(R) \oplus \mathsf{encode}(J)) \\ C_B = \quad & (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4}) = \quad (h^s, S \cdot e(g,h)^{sb_1}, h^{sm_2b_2}, F(S) \oplus \mathsf{encode}(J)) \end{aligned}$$

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

$$X := c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})} = S \cdot e(g,h)^{sb_1} \cdot \frac{e(g^r, h^{sm_2b_2})}{e(g^{ra_2m_2+m_1}, h^s)} = S \cdot e(g,h)^{s(b_1-m_1)} \cdot (e(g,h)^{sm_2(b_2-a_2)})^r$$

Now, D will return 1 if and only if $sk_A = \mathsf{decode}((F(S) \oplus \mathsf{encode}(J)) \oplus F(X))$. What is the probability that this event occurs? First, suppose that $sm_2(b_2 - a_2) \bmod p \neq 0$ (event E_3), which happens with probability $\geq 1 - 3/(p-1) = (p-4)/(p-1)$ for honest executions. Next, consider the values J, 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 $F(X) = F(S) \oplus \mathsf{encode}(J) \oplus \mathsf{encode}(sk_A)$? First, we observe that since $sm_2(b_2 - a_2) \neq 0$ and r is chosen uniformly at random in $\mathbb{Z}_p$, then X is also distributed uniformly at random in $\mathbb{G}_T$. The question reduces to: for a fixed value $K \in \{0,1\}^{\ell(\lambda)}$, a pseudorandom generator F and a random seed X, what is the chance that $K = F(X)$? Since, by assumption, F is computationally indistinguishable from a uniform, random function, then this probability can be at most $2^{-\ell(\lambda)}$ plus a negligible amount $\nu(\lambda)$, where λ is the security parameter.

Thus, D's total probability of success, when it does not abort, is:

$$\begin{aligned} \Pr[D \text{ wins}] &= \Pr[E_1] \cdot \Pr[D \text{ wins } | E_1] + \Pr[\bar{E}_1] \cdot \Pr[D \text{ wins } | \bar{E}_1] \\ &= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \left(\Pr[E_2] \cdot \Pr[D \text{ wins } | E_2] + \Pr[\bar{E}_2] \cdot \Pr[D \text{ wins } | \bar{E}_2] \right) \\ &\geq \frac{1}{4} + \frac{1}{2} \cdot \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \left(\Pr[E_3] \cdot \Pr[D \text{ wins } | E_3] \right) \right) \\ &\geq \frac{1}{2} + \frac{1}{4} \cdot \left(\frac{p-4}{p-1} \cdot (1 - 2^{-\ell(\lambda)} - \nu(\lambda)) \right) \\ &\geq \frac{5}{8} - \frac{(2^{-\ell(\lambda)} + \nu(\lambda))}{4} \quad \text{for all } p \geq 7 \end{aligned}$$

Of course, for practical 80-bit or higher values of p, this probability is much closer to 3/4. □

4 Extension: A Counterexample for CCA Security

We now 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 [34] and refined by Dolev, Dwork and Naor [17] and Sahai [36]. Specifically, our building blocks will be:

1. The IND-CPA-secure cryptosystem $\Pi_{\mathsf{cpa}} = (G, E, D)$ from Section 3. Let $E(pk, m; r)$ denote the encryption of message m under public key pk with randomness r.

2. An adaptively non-malleable (or simulation-sound) non-interactive zero-knowledge (NIZK) proof system $\Gamma = (P, V, S = (S_1, S_2))$ with unpredictable simulated proofs and uniquely applicable proofs for the language L of consistent pairs of encryptions, defined as:

$$L = \{(e_0, e_1, c_0, c_1) : \exists m, r_0, r_1 \in \{0,1\}^* \text{ s.t. } c_0 = E(e_0, m; r_0) \text{ and } c_1 = E(e_1, m; r_1)\}.$$

See [15] and [36] to show that such a proof system for L can be realized under relatively mild assumptions, such as the difficulty of factoring Blum integers. Alternatively, Groth [24] and Camenisch et al. [12] realize simulation-sound NIZK proof systems for certain bilinear group statements, which might apply here, although the one non-triviality would be handling the PRG.

Construction Π_{cca}. The construction $\Pi_{\mathsf{cca}} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, following [36] 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 4.1 *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 3.1 and [36], Theorem 4.1.)

Next, we have a more surprising result. It is just as easy to break this CCA system as it was to break the CPA system, under a circular attack.

Theorem 4.2 *Encryption scheme Π_{cpa} is not* IND-CIRC-CPA *secure for cycles of length 2.*

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 3.9, 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 succeed with probability almost $5/8$ as before. □

5 Conclusion and Open Problems

In this work, we presented new public-key encryption systems that are secure in the IND-CPA and IND-CCA sense, but fail catastrophically in the presence of a 2-cycle. Together with the IND-CPA result of Acar et al. [2], this answers a longstanding, foundational question on whether standard

definitions of encryption capture circular security, for cycles larger than self-loops. Our constructions are quite practical and not obviously flawed (in a circular sense); indeed, had they been proposed in a different context we believe that their weakness might have been overlooked. Given these cautionary, negative results, the search for new and practical circular-secure systems becomes all the more interesting. It is also of increased importance that encryption systems employed in scenarios where key cycles may intentionally or accidentally occur be replaced or further analyzed for weaknesses. Our work leaves open the interesting problems of finding a counterexample for cycles of arbitrary size, under different complexity assumptions or relative to only the assumption that IND-CPA-secure systems exist.

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 *To appear in EUROCRYPT '10*, volume LNCS. Springer-Verlag, 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. ACM Press, 2005.

[7] 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`.

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

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

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

[11] 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.

[12] 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.

[13] 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.

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

[15] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Necessary and sufficient assumptions for non-iterative zero-knowledge proofs of knowledge for all NP relations. In *ICALP '00*, volume 1853 of LNCS, pages 451–462, 2000.

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

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

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

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

[20] E. Ghadafi, N.P. Smart, and B. Warinschi. Groth-Sahai proofs revisited, 2009. Cryptology ePrint Archive: Report 2009/599.

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

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

[23] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT '08*, volume 5350 of LNCS, pages 179–197, 2008.

[24] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT '06*, volume 4284 of LNCS, pages 444–459. Springer, 2006.

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

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

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

[28] 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.

[29] 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.

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

[31] Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In *International Conference on Information Security and Cryptology (ICISC)*, volume 2971 of LNCS, pages 55–66, 2003.

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

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

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

[35] 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.

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

[37] 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 An Alternative Counterexample for CPA Security

As mentioned in Section 3, 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 (though specific implementations may always do so for unexpected reasons).

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 $\mathsf{IND\text{-}CPA}$-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 3 since it permits an adversary to win the* $\mathsf{IND\text{-}CIRC\text{-}CPA}$ *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 $\Pi'_{\mathsf{cpa}} = (\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ uses $\Pi_{\mathsf{cpa}} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ 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 $pk_2 = (1, \cdot, \cdot)$.[4] 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:

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

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

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

Correctness and IND-CPA Security. Correctness follows trivially from the correctness of Π_{cpa}.

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

We omit a formal argument for CPA security, but observe that as the keypairs (pk_1, sk_1) and (pk_2, sk_2) are independently generated then, under a standard hybrid argument, the encryption $C = (\mathsf{Encrypt}(pk_1, m_2), \mathsf{Encrypt}(pk_2, m_1))$ is an IND-CPA-secure encryption of (m_1, m_2) given Theorem 3.1.

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

Theorem A.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 3.4. As in that attack, we assume that the adversary receives $C_A = \mathsf{Encrypt}(pk_A, sk_B)$ and $C_B = \mathsf{Encrypt}(pk_B, sk_A)$ 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:

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

[4] 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.

If we are in case 1, then we simply apply the exact attack from Section 3.4 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:

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

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

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

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. □