Circular-Secure Encryption from Decision Diffie-Hellman

Dan Boneh[1]*, Shai Halevi[2], Mike Hamburg[1], and Rafail Ostrovsky[3]**

[1] Computer Science Dept., Stanford University — {dabo,mhamburg}@cs.stanford.edu
[2] IBM Research — shaih@alum.mit.edu
[3] Computer Science Dept. and Dept. of Math., UCLA — rafail@cs.ucla.edu

Abstract. We describe a public-key encryption system that remains secure even encrypting messages that depend on the secret keys in use. In particular, it remains secure under a "key cycle" usage, where we have a cycle of public/secret key-pairs $(\mathrm{pk}_i, \mathrm{sk}_i)$ for $i = 1, \ldots, n$, and we encrypt each sk_i under $\mathrm{pk}_{(i \bmod n)+1}$. Such usage scenarios sometimes arise in key-management systems and in the context of anonymous credential systems. Also, security against key cycles plays a role when relating "axiomatic security" of protocols that use encryption to the "computational security" of concrete instantiations of these protocols.

The existence of encryption systems that are secure in the presence of key cycles was wide open until now: on the one hand we had no constructions that provably meet this notion of security (except by relying on the random-oracle heuristic); on the other hand we had no examples of secure encryption systems that become demonstrably insecure in the presence of key-cycles of length greater than one.

Here we construct an encryption system that is circular-secure against chosen-plaintext attacks under the Decision Diffie-Hellman assumption (without relying on random oracles). Our proof of security holds even if the adversary obtains an encryption clique, that is, encryptions of sk_i under pk_j for all $1 \le i, j \le n$. We also construct a circular counterexample: a one-way secure encryption scheme that breaks completely if an encryption cycle (of any size) is published.

1 Introduction

Secure encryption is arguably the most basic task in cryptography, and significant work has gone into defining and attaining it. All commonly accepted definitions for secure encryption [2, 3, 8, 10, 16, 17, 19] assume that the plaintext messages to be encrypted cannot depend on the secret decryption keys themselves. The danger of encrypting messages that the adversary cannot find on its own was already noted more than two decades ago by Goldwasser and Micali [10, §5.1].

Over the last few years, however, it was observed that in some situations the plaintext messages do depend on the secret keys. An important example is when we have a cycle of public/secret key-pairs $(\mathrm{pk}_i, \mathrm{sk}_i)$ for $i = 1, \ldots, n$, and we encrypt each sk_i under $\mathrm{pk}_{(i \bmod n)+1}$. Security in this more demanding setting was termed *key-dependent message security* (KDM-security) by Black et al. [4] and *circular security* by Camenisch and Lysyanskaya [6].

Such situations may arise due to careless key management, for example a backup system may store the backup encryption key on disk and then encrypt the entire disk, including the key, and backup the result. Another example is the BitLocker disk encryption utility (used

* Supported by NSF and the Packard Foundation.

** Partially supported by IBM Faculty Award, Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389 and U.C. MICRO grant.

in Windows Vista) where the disk encryption key can end up on disk and be encrypted along with the disk contents. There are also situations where circular security is needed "by design", e.g., Camenisch and Lysyanskaya used it in their anonymous credential system [6] to discourage users from delegating their secret keys. Finally, in the formal-methods community the notion of key-dependent security from [4] was used to prove equivalence between "computational security" and "axiomatic security" [1, 18].

Definitions of security for this setting were given by Black et al. [4], who defined models of KDM security in both the symmetric and public-key settings. In their public-key model the adversary is given public keys $\mathrm{pk}_1, \ldots, \mathrm{pk}_n$ and can access an oracle $\mathcal{O}$ that returns the encryption of $g(\mathrm{sk}_1, \ldots, \mathrm{sk}_n)$ under pk_i for any polynomial-time function g and any index $1 \leq i \leq n$ of the adversary's choosing. (A key-cycle can be obtained in this model when the adversary requests the encryption of sk_i under $\mathrm{pk}_{(i \bmod n)+1}$ for all i.) The system is KDM-secure if the adversary cannot distinguish the oracle $\mathcal{O}$ from an oracle that always returns an encryption of (say) the all-zero string.

A simple example of KDM is when an encryption system is used to encrypt its own secret key (i.e., a cycle of size one). It is straightforward to construct a secure encryption scheme that becomes completely insecure once the adversary sees such self-referential ciphertext, and similarly it is straightforward to construct an encryption scheme that remains secure under such self-referential encryption [4].[1] The question becomes much harder when dealing with more complicated key-dependent messages, for example key-cycles of size more than one. For these cases, the problem has been wide open. On one hand, we had no examples of encryption systems that are secure without key-cycles but demonstrably insecure in the presence of a key cycle of size more than one. On the other hand, we had no constructions that can be proved to meet such notions of security (except by relying on the random-oracle heuristic). Some initial steps toward constructions in the standard model are given in [12] (who focused on other primitives such as PRFs) and [15] (who achieved weaker variants of these security notions).

1.1 Our results

Our main result is a public-key system that is circular-secure (or even "clique-secure") in the standard model under the Decision Diffie-Hellman assumption. That is, even an adversary who sees an encryption of sk_i under pk_j for all $1 \leq i, j \leq n$ cannot distinguish the ciphertexts from n^2 encryptions of (say) the all-zero string. In fact, we prove a slightly stronger result by showing that our system is KDM-secure against chosen-plaintext attacks in the model of Black et al. [4], when the adversary is restricted to affine functions of the keys. Hence, our system tolerates the adversary seeing encryption cliques (or even encryptions of more complicated functions of the secret keys) without compromising security.

[1] For the former, start from a secure encryption system (where secret keys are not valid ciphertexts), and modify the encryption algorithm so that when encrypting the secret key it outputs it in the clear. For the latter, modify the encryption algorithm so that when encrypting the secret key it outputs the encryption of a distinguished symbol $\perp$.

The difficulty in constructing such a system is the simulation of an encryption clique without knowledge of any of the secret keys. We overcome this difficulty by having a system which is sufficiently homomorphic that such a clique can be constructed directly. We point out that one may be tempted to use a Cramer-Shoup-like construction and simulation [7] to prove n-circular security. After all, a Cramer-Shoup simulator is in possession of all secret keys (needed for responding to decryption queries) and can use them to create an encryption clique to give to the adversary. Unfortunately, we could not get this intuition to work. The problem is that the simulator has to embed the DDH challenge into the circular clique, but it is difficult to do so while creating a valid clique.

In Section 5 we also take a first step toward showing that standard security notions do not imply circular security. Specifically, we construct a very simple one-way encryption system that breaks completely as soon as a key-cycle of any size is published.

2 KDM security: definitions and properties

We begin by reviewing the definitions of Key-Dependent Message security (KDM) in the public-key setting from Black et al. [4]. We use a small extension of the definition, used also in [12], that restricts the adversary to a particular set of functions.

A public-key encryption system $\mathcal{E}$ consists of three algorithms $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ where G is a key-generation algorithm that takes as input a security parameter λ and outputs a public/secret key pair $(\mathrm{pk}, \mathrm{sk})$; $\mathsf{E}(\mathrm{pk}, m)$ encrypts message m with public key pk; and $\mathsf{D}(\mathrm{sk}, c)$ decrypts ciphertext c with secret key sk. We have the usual correctness condition, asserting that decryption correctly recovers the plaintext message from the ciphertext (with probability one).

We use S to denote the space of secret keys output by $\mathsf{G}()$ and use M to denote the message (plaintext) space. Throughout the paper we assume that $S \subseteq M$ so that any secret key sk can be encrypted using any public key pk'. All of these notations assume an implied security parameter λ.

2.1 KDM security with respect to a set of functions $\mathcal{C}$

Informally, KDM security implies that the adversary cannot distinguish the encryption of a key-dependent message from an encryption of 0. We define key-dependence relative to a fixed set of functions $\mathcal{C}$.[2] Let $n > 0$ be an integer and let $\mathcal{C}$ be a finite set of functions $\mathcal{C} := \{f : S^n \to M\}$. For each function $f \in \mathcal{C}$ we require that $|f(z)|$ is the same for all inputs $z \in S^n$ (i.e. the output length is independent of the input).

We define KDM security with respect to $\mathcal{C}$ using the following game that takes place between a challenger and an adversary $\mathcal{A}$. For an integer $n > 0$ and a security parameter λ the game proceeds as follows:

init. The challenger chooses a random bit $b \xleftarrow{R} \{0, 1\}$. It generates $(\mathrm{pk}_1, \mathrm{sk}_1), \ldots, (\mathrm{pk}_n, \mathrm{sk}_n)$ by running $\mathsf{G}(\lambda)$ n times, and sends the vector $(\mathrm{pk}_1, \ldots, \mathrm{pk}_n)$ to $\mathcal{A}$.

[2] Technically $\mathcal{C}$ is a family of sets, parameterized by the security parameter.

queries. The adversary repeatedly issues queries where each query is of the form (i, f) with $1 \le i \le n$ and $f \in \mathcal{C}$. The challenger responds by setting

$$y \leftarrow f(\mathrm{sk}_1, \ldots, \mathrm{sk}_n) \in M \quad \text{and} \quad c \xleftarrow{R} \begin{cases} \mathsf{E}(\mathrm{pk}_i,\ y) & \text{if } b = 0 \\ \mathsf{E}(\mathrm{pk}_i,\ 0^{|y|}) & \text{if } b = 1 \end{cases}$$

and sends c to $\mathcal{A}$.

finish. Finally, the adversary outputs a bit $b' \in \{0, 1\}$.

We say that $\mathcal{A}$ is a $\mathcal{C}$-KDM adversary and that $\mathcal{A}$ wins the game if $b = b'$. Let W be the event that $\mathcal{A}$ wins the game and define $\mathcal{A}$'s advantage as

$$\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}](\lambda) \ := \ \left| \Pr[W] - \frac{1}{2} \right|$$

Definition 1. *We say that a public-key encryption scheme $\mathcal{E}$ is n-***way KDM-secure with respect to** $\mathcal{C}$ *if* $\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}](\lambda)$ *is a negligible function of λ for any adversary $\mathcal{A}$ that runs in expected polynomial time in λ.*

We are primarily interested in function classes $\mathcal{C}$ that imply that the public-key system $\mathcal{E}$ is circular secure. Specifically, we look for function classes $\mathcal{C} := \{f : S^n \to M\}$ that are non-trivial, in the sense that they contain:

- all $|M|$ constant functions $f : S^n \to M$ (recall that a constant function maps all inputs in S^n to some constant $m \in M$), and
- all n selector functions $f_i(x_1, \ldots, x_n) = x_i$ for $1 \le i \le n$.

It is easy to see that KDM-security with respect to such non-trivial function class implies standard semantic security (even for symmetric encryption), since the constant functions let the adversary obtain the encryption of any message of its choice. The selector functions imply *circular security* since they let the adversary obtain $\mathsf{E}(\mathrm{pk}_i,\ \mathrm{sk}_j)$ for all $1 \le i, j \le n$.

The main result in this paper is a public-key system that is KDM-secure relative to a non-trivial function class (and hence also circular-secure). Specifically, we prove security relative to the class of *affine functions* (over the group that is used in the system).

2.2 Decision Diffie-Hellman

Let $\mathbb{G}$ be a group of prime order q. We let $\mathcal{P}_{\mathrm{DDH}}$ be the distribution (g, g^x, g^y, g^{xy}) in $\mathbb{G}^4$ where g is a random generator of $\mathbb{G}$ and x, y are uniform in $\mathbb{Z}_q$. We let $\mathcal{R}_{\mathrm{DDH}}$ be the distribution (g, g^x, g^y, g^z), where g is a random generator of $\mathbb{G}$ and x, y, z are uniform in $\mathbb{Z}_q$ subject to $z \ne xy$. A DDH adversary $\mathcal{A}$ takes as input a tuple (g, h, u, v) in $\mathbb{G}^4$ and outputs 0 or 1. Define

$$\mathrm{DDH\,Adv}[\mathcal{A}, \mathbb{G}] := \left| \Pr[x \xleftarrow{R} \mathcal{P}_{\mathrm{DDH}} \ : \ \mathcal{A}(x) = 1] \ - \ \Pr[x \xleftarrow{R} \mathcal{R}_{\mathrm{DDH}} \ : \ \mathcal{A}(x) = 1] \right|$$

Informally, we say that DDH holds in $\mathbb{G}$ if $\mathrm{DDH\,Adv}[\mathcal{A}, \mathbb{G}]$ is negligible for all efficient $\mathcal{A}$.

3 A circular-secure encryption scheme

We build a circular-secure encryption system (for any n) based on the Decision Diffie-Hellman (DDH) assumption. The system is a generalization of the ElGamal system where the secret key is a bit vector rather than an element in $\mathbb{Z}_q$. Let $\mathbb{G}$ be a group of prime order q and g a fixed generator of $\mathbb{G}$. The size of $\mathbb{G}$ is determined by a security parameter λ, in particular, $1/q$ is negligible in λ.

The public-key encryption system $\mathcal{E}$:

- **Key Generation.** Let $\ell := \lceil 3 \log_2 q \rceil$. Choose random $g_1, \ldots, g_\ell$ in $\mathbb{G}$ and a random vector $\boldsymbol{s} = (s_1, \ldots, s_\ell)$ in $\{0,1\}^\ell$. Let $h \leftarrow (g_1^{s_1} \cdots g_\ell^{s_\ell})^{-1}$ and define the public and secret keys to be
$$\mathrm{pk} := (g_1, \ldots, g_\ell, h) \quad \text{and} \quad \mathrm{sk} := (g^{s_1}, \ldots, g^{s_\ell})$$
Note that the secret key sk is a random vector $\boldsymbol{s}$ in $\{0,1\}^\ell$ encoded as a vector of ℓ group elements.
- **Encryption.** To encrypt a group element $m \in \mathbb{G}$, choose a random $r \overset{R}{\leftarrow} \mathbb{Z}_q$ and output the ciphertext
$$\left(\ g_1^r,\ \ldots,\ g_\ell^r,\ \ h^r \cdot m\ \right)$$
- **Decryption.** Let $(c_1, \ldots, c_\ell,\ d)$ be a ciphertext and $\mathrm{sk} = (v_1, \ldots, v_\ell)$ a secret key. Do:
  - decode the secret key: for $i = 1, \ldots, \ell$ set $s_i \leftarrow 0$ if $v_i = 1$ and $s_i \leftarrow 1$ otherwise;
  - output $m \leftarrow d \cdot (c_1^{s_1} \cdots c_\ell^{s_\ell})$.

It is easy to verify that the system is correct, that is, the decryption algorithm decrypts properly constructed ciphertexts.

3.1 Discussion and outline of the security proof

Proving that the system is circular secure is somewhat involved. Before proving security, we give some intuition for the construction and its proof. First, consider the basic ElGamal system. The public key is a pair $(g, g^x) \in \mathbb{G}^2$ and the secret key is $x \in \mathbb{Z}_q$. A 1-cycle for this system, namely $\mathsf{E}(\mathrm{pk}, \mathrm{sk})$, is a ciphertext $(g^r,\ e(x) \cdot g^{rx})$ where $e(\cdot)$ is some invertible encoding function mapping $\mathbb{Z}_q$ to $\mathbb{G}$. To prove 1-circular security we would need to show that the 4-tuple
$$\left(g,\ g^x,\ g^r,\ e(x) \cdot g^{rx}\right) \quad \in \mathbb{G}^4 \tag{1}$$
is indistinguishable from a random 4-tuple in $\mathbb{G}^4$, but this is unlikely to follow from DDH.

It is tempting to define the secret key as $\mathrm{sk} := v^x$ (for some generator v), in which case the 1-cycle becomes $(g^r,\ v^x \cdot g^{rx})$. The resulting system can be shown to be 1-circular secure under DDH. Unfortunately, the system does not work since one cannot decrypt ElGamal encryptions using the key $\mathrm{sk} = v^x$.

As a compromise between these two systems, we pick the secret key as an ℓ-bit vector $\boldsymbol{s} \overset{R}{\leftarrow} \{0,1\}^\ell$ and store the key as $\mathrm{sk} := (g^{s_1}, \ldots, g^{s_\ell})$. We decrypt using sk by going back to the bit-vector representation. The challenge is to prove n-circular security from the DDH assumption.

Proof outline It is instructive to attempt a direct proof that the system $\mathcal{E}$ is 1-circular secure. Observe that in the system $\mathcal{E}$, it is easy to construct "ciphertext vectors" whose decryption are elements of the secret key: For every $1 \leq i \leq \ell$, the $(\ell+1)$-vector

$$(1, \ldots, 1, g, 1 \ldots, 1)$$

with g in position i and 1's everywhere else, decrypts to the secret-key element g^{s_i}. Hence the simulator can generate "an encryption of the secret key" without knowing the secret key itself. This almost suffices for the proof, except that these vectors are not really valid ciphertext vectors, since the encryption algorithm would never output them.

We therefore begin by moving to an "expanded variant" of our system (which we call $\mathcal{E}_1$) that has the same decryption procedure, but where every $(\ell+1)$-vector is a valid ciphertext (see description later in Section 3.2). Moreover, $\mathcal{E}_1$ has the same "blinding" properties as ElGamal, so the simulator can produce not just one encryption of the secret key but *a random encryption* of it. This is enough to prove that the expanded system $\mathcal{E}_1$ is 1-circular secure. Moving from 1-circular to n-circular security is done using homomorphic properties: $\mathcal{E}_1$ is homomorphic with respect to both the plaintext and secret key, so it is possible to translate an encryption of m with respect to secret key $\boldsymbol{s}$ to an encryption of $m \cdot d$ with respect to secret key $\boldsymbol{s} \oplus \boldsymbol{\delta}$, just by knowing d and $\boldsymbol{\delta}$. This allows the simulator to first produce a 1-cycle and then expand it into an n-clique. The circular security (or even clique-security) of $\mathcal{E}_1$ follows.

Finally we deduce the security of $\mathcal{E}$ from that of $\mathcal{E}_1$, roughly because the adversary gets "strictly less information" when attacking $\mathcal{E}$ than when attacking the expanded variant $\mathcal{E}_1$.

3.2 Proof of circular-security

We now prove that the system $\mathcal{E}$ provides circular security. As mentioned above, we actually prove a slightly stronger statement, namely that $\mathcal{E}$ is KDM-secure with respect to the set of affine functions.

Affine functions. The set of affine functions acting on S^n is defined as follows. Let $\mathrm{sk}_1, \ldots, \mathrm{sk}_n$ be n secret keys generated by G (each an ℓ-vector over $\mathbb{G}$). Let $\boldsymbol{s}$ be the vector in $\mathbb{G}^{n\ell}$ obtained by concatenating these n secret keys. For every $n\ell$-vector $\boldsymbol{u} = (u_i)$ over $\mathbb{Z}_q$ and every scalar $h \in \mathbb{G}$, there is a natural map from $\mathbb{G}^{n\ell}$ to $\mathbb{G}$, that can be informally described as $f_{\boldsymbol{u},h} : \quad \boldsymbol{s} \longrightarrow (\boldsymbol{u} \cdot \boldsymbol{s} + h)$. More precisely, we have

$$f_{\boldsymbol{u},h}(\boldsymbol{s}) \stackrel{\text{def}}{=} \prod_{i=1}^{n\ell} s_i^{u_i} \cdot h \quad \in \mathbb{G}.$$

We call $f_{\boldsymbol{u},h}$ an affine function from $\mathbb{G}^{n\ell}$ to $\mathbb{G}$.

Definition 2. *The set of affine functions $\mathcal{C}_{n\ell}$ is the set of all functions $f_{\boldsymbol{u},h}$, where $\boldsymbol{u} \in \mathbb{Z}_q^{n\ell}$ and $h \in \mathbb{G}$.*

The set $\mathcal{C}_{n\ell}$ acts on n-tuples of secret keys by viewing the n-tuple as a vector in $\mathbb{G}^{n\ell}$, and it maps every such vector to an element of $\mathbb{G}$.

KDM-security theorem with respect to $\mathcal{C}_{n\ell}$. The following theorem shows that if the DDH assumption holds in the group $\mathbb{G}$ then $\mathcal{E}$ is n-way KDM-secure with respect to the set $\mathcal{C}_{n\ell}$ of affine functions, for any $n = n(\lambda)$ that is polynomial in the security parameter. Circular security follows since $\mathcal{C}_{n\ell}$ contains the constant and selector functions.[3]

Theorem 1. *For any $n > 0$ and for any $\mathcal{C}_{n\ell}$-KDM adversary $\mathcal{A}$, there exists a DDH adversary $\mathcal{B}$ (whose running time is about the same as that of $\mathcal{A}$) such that*

$$\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}] \leq (3\ell - 2) \cdot \mathrm{DDH}\,\mathrm{Adv}[\mathcal{B}, \mathbb{G}] + 1/q$$

Note that this bound is independent of n.

Switching to additive notation Our proof requires a fair amount of linear algebra. To make it clearer, we will be using additive notation for the group $\mathbb{G}$; note that $\mathbb{G}$ and $\mathbb{G}^k$ are vector spaces over $\mathbb{Z}_q$. (Recall that we already used additive notation when we informally described the class of affine functions above.) To avoid ambiguity, we use Latin letters for elements of $\mathbb{Z}_q$ and Greek letters for elements of $\mathbb{G}$. In particular, let $\mathbb{G}$ be generated by γ. We use lower-case letters for scalars and column vectors, and upper-case letters for matrices. We write $\boldsymbol{\mu} \cdot \boldsymbol{v}$ for the inner product and $\boldsymbol{\mu} \times \boldsymbol{v}$ for the outer product:

$$\boldsymbol{\mu} \cdot \boldsymbol{v} := \boldsymbol{\mu}^\top \boldsymbol{v} = \sum_i \mu_i v_i \qquad \text{and} \qquad \boldsymbol{\mu} \times \boldsymbol{v} := \boldsymbol{\mu}\, \boldsymbol{v}^\top = (\mu_i v_j)_{i,j} \in \mathbb{G}^{\dim(\mu) \times \dim(v)}$$

When we write the product of a matrix or vector of elements of $\mathbb{G}$ with a matrix or vector of elements of $\mathbb{Z}_q$, we mean to use the standard formula. For example, by $\boldsymbol{\mu} \cdot \boldsymbol{v}$ we mean $\sum_i \mu_i v_i$, which would be written in multiplicative notation as $\prod_i \mu_i^{v_i}$. It is easily seen that the usual properties of vectors and matrices still hold in this notation (for any term involving at most one literal from $\mathbb{G}$ and all other literals from $\mathbb{Z}_q$).

We use 0 to denote both the number zero and the identity element in $\mathbb{G}$, the meaning will be clear from the context. We write 0^ℓ for a column vector of ℓ zeros, and $0^{k \times \ell}$ for a $k \times \ell$ matrix of zeros. We write Id_i for the identity matrix in $\mathbb{Z}_q^{i \times i}$. When A, B are two matrices with the same number of rows (and both over $\mathbb{G}$ or both over $\mathbb{Z}_q$), then we write $\left(A \middle| B\right)$ for the augmented matrix consisting of all the columns of A followed by all the columns of B.

The definitions of linear independence of vectors, vector spaces and subspaces, rank of matrices, etc., are all standard, and we use the same notions for both $\mathbb{G}$ and $\mathbb{Z}_q$. We write $\mathrm{Rk}_i\left(\mathbb{Z}_q^{a \times b}\right)$ (resp $\mathrm{Rk}_i\left(\mathbb{G}^{a \times b}\right)$) for the set of matrices in $\mathbb{Z}_q^{a \times b}$ (resp $\mathbb{G}^{a \times b}$) with rank i. As a special case, we write $\mathrm{GL}_i(\mathbb{Z}_q)$ for the invertible $i \times i$ matrices over $\mathbb{Z}_q$.

The system $\mathcal{E}$ in additive notation

[3] The set $\mathcal{C}_{n\ell}$ includes "selector functions" for each element from each secret key (rather than "selector functions" that select entire keys). This makes no difference in our case, since our scheme encrypts element by element (so a secret key is encrypted as ℓ separate ciphertext vectors, one for each element of the secret key).

- **Key Generation.** Let $\ell := \lceil 3 \log_2 q \rceil$. Choose a random nonzero vector $\boldsymbol{\psi} \xleftarrow{R} \mathbb{G}^\ell$ and a random vector $\boldsymbol{s}$ in $\{0,1\}^\ell \subset \mathbb{Z}_q^\ell$. Let $\delta \leftarrow -\boldsymbol{\psi} \cdot \boldsymbol{s} \in \mathbb{G}$ and define the public and secret keys to be
$$\mathrm{pk} := \left(\boldsymbol{\psi}^\top | \delta\right) \in \mathbb{G}^{1\times(\ell+1)} \qquad \text{and} \qquad \mathrm{sk} := \boldsymbol{s}\gamma \in \mathbb{G}^\ell$$
Though the secret key is encoded as $\mathrm{sk} = \boldsymbol{s}\gamma \in \mathbb{G}^\ell$, below it will be convenient to consider also the decoded form. Specifically, we refer to the $\ell+1$ binary vector $\boldsymbol{s'} = \left(\boldsymbol{s}^\top | 1\right)^\top \in \mathbb{Z}_q^{\ell+1}$ as the *decoded secret key.*
- **Encryption.** To encrypt a message $\mu \in \mathbb{G}$, choose a random $r \xleftarrow{R} \mathbb{Z}_q^n$ and output the ciphertext row-vector
$$\boldsymbol{\xi}^\top \leftarrow \left(r\boldsymbol{\psi}^\top | r\delta + \mu\right) = r\,\mathrm{pk} + \left(0^{1\times\ell} | \mu\right) \quad \in \mathbb{G}^{1\times(\ell+1)} \tag{2}$$
- **Decryption.** Let $\boldsymbol{\xi}^\top \in \mathbb{G}^{1\times(\ell+1)}$ be the ciphertext. Decryption is just an inner product between the ciphertext and the decoded secret key:
$$\mu \leftarrow \boldsymbol{\xi} \cdot \left(\boldsymbol{s}^\top | 1\right)^\top$$
Decryption works since the decoded secret key $\left(\boldsymbol{s}^\top | 1\right)^\top$ is orthogonal to the public key pk.

We observe that an $\ell + 1$ vector over $\mathbb{G}$ is decrypted to zero if and only if it belongs to the subspace orthogonal to the decoded secret key $\boldsymbol{s'}$, and every coset of this subspace is decrypted to a different element of $\mathbb{G}$. On the other hand, "valid encryptions of zero" (i.e., the ones obtained from the encryption algorithm) are taken from a small subspace of the vectors orthogonal to $\boldsymbol{s'}$, namely the one-dimensional subspace spanned by the public key pk. Similarly, "valid encryptions" of other elements are obtained as shifting this one-dimensional subspace by multiples of $\left(0^{1\times\ell} | 1\right)$.

A few lemmata We present some simple lemmata and facts about $\mathcal{E}$. First, we show that DDH implies that it is difficult to determine the rank of a matrix of group elements. In particular, it is difficult to distinguish a random matrix of rank r_1 from a random matrix of rank $r_2 > r_1$.

Lemma 1 (Matrix DDH). *Let $1 \le r_1 < r_2 \le a, b$ be positive integers, and let $\mathcal{A} : \mathbb{G}^{a\times b} \to \{0,1\}$ be a polynomial-time algorithm. Write*
$$P(\mathcal{A}, i) := \Pr\left[\Phi \xleftarrow{R} \mathrm{Rk}_i\left(\mathbb{G}^{a\times b}\right) \ : \ \mathcal{A}(\Phi) = 1\right]$$
Then there is a DDH adversary $\mathcal{B}$, running in about the same time as $\mathcal{A}$, such that
$$\left|P(\mathcal{A}, r_2) - P(\mathcal{A}, r_1)\right| \le (r_2 - r_1)\ \mathrm{DDH\,Adv}[\mathcal{B}, \mathbb{G}]$$

Proof. We use a hybrid argument between the $r_2 - r_1 + 1$ distributions
$$\mathrm{Rk}_i\left(\mathbb{G}^{a\times b}\right) \quad \text{where} \quad i \in [r_1, r_2]$$

The algorithm $\mathcal{B}$ is given a DDH challenge $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$. It picks a random $i \overset{R}{\leftarrow} [r_1 + 1, r_2]$ and sets

$$\Phi_1 := \left(\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & & \\ \hline \alpha_3 & \alpha_4 & & \\ \hline & & \gamma\,\mathrm{Id}_{i-2} & \\ \hline & & & 0^{(a-i)\times(b-i)} \end{array}\right) \in \mathbb{G}^{a\times b}$$

with all the other blocks zero. $\mathcal{B}$ then chooses

$$L \overset{R}{\leftarrow} \mathrm{GL}_a(\mathbb{Z}_q) \ \text{ and } \ R \overset{R}{\leftarrow} \mathrm{GL}_b(\mathbb{Z}_q) \quad \text{and sets} \quad \Phi_2 := L\,\Phi_1\,R$$

$\mathcal{B}$ now calls $\mathcal{A}(\Phi_2)$ and outputs whatever $\mathcal{A}$ outputs.

Now if $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ was drawn from $\mathcal{P}_{\text{DDH}}$, then Φ_1 has rank $i-1$, and Φ_2 is uniform in $\mathrm{Rk}_{i-1}\left(\mathbb{G}^{a\times b}\right)$. But if $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ was drawn from $\mathcal{R}_{\text{DDH}}$, then Φ_1 has rank i, and Φ_2 is uniform in $\mathrm{Rk}_i\left(\mathbb{G}^{a\times b}\right)$. The lemma then follows by the standard hybrid argument. □

We will also need the following lemma on universal hashing. Recall that a distribution $\mathcal{D}$ on a set $\mathcal{X}$ is ϵ-uniform if $\sum_{x\in X} \left|\mathcal{D}(x) - \frac{1}{|\mathcal{X}|}\right| \leq \epsilon$.

Lemma 2 (Simplified left-over hash lemma). *Let $\mathcal{H}$ be a 2-universal hash family from a set $\mathcal{X}$ to a set $\mathcal{Y}$. Then the distribution*

$$(H, H(x)) \ \ \textit{where} \ \ H \overset{R}{\leftarrow} \mathcal{H} \ \ \textit{and} \ \ x \overset{R}{\leftarrow} \mathcal{X}$$

is $\sqrt{\frac{|\mathcal{Y}|}{4|\mathcal{X}|}}$-uniform on $\mathcal{H} \times \mathcal{Y}$.

Proof. This is an immediate corollary from [13] (see also [21, Theorem 6.21]). □

Recall that the secret key in our system is a vector in $\{0,1\}^\ell$ where $\ell = \lceil 3\log_2 q \rceil$. We therefore obtain the following corollary of Lemma 2.

Corollary 1. *Let $\boldsymbol{r} \overset{R}{\leftarrow} \mathbb{Z}_q^\ell$, and $\boldsymbol{s} \overset{R}{\leftarrow} \{0,1\}^\ell$. Then $\left(\boldsymbol{r}^\top | {-\boldsymbol{r}\cdot\boldsymbol{s}}\right)^\top$ is $\frac{1}{q}$-uniform in $\mathbb{Z}_q^{n+1}$.*

Proof. Let $H_{\boldsymbol{r}}(\boldsymbol{s}) := -\boldsymbol{r}\cdot\boldsymbol{s}$. Then $\{H_{\boldsymbol{r}} : \boldsymbol{r} \in \mathbb{Z}_q^\ell\}$ is 2-universal, so that $\left(\boldsymbol{r}^\top | {-\boldsymbol{r}\cdot\boldsymbol{s}}\right)^\top$ is $\sqrt{\frac{q}{4\cdot 2^\ell}}$-uniform in $\mathbb{Z}_q^{n+1}$. Since $\ell = \lceil 3\log_2 q \rceil$ we have $\sqrt{\frac{q}{4\cdot 2^\ell}} \leq \frac{1}{2q} < \frac{1}{q}$. □

We note that Erdös and Hall [9] proved a slightly stronger version of Corollary 1 — they obtain a similar result with a smaller ℓ (i.e. $\ell \approx \lceil 2\log_2 q \rceil$). This enables us to slightly shorten our public and secret keys. However, the proof of Corollary 1 using the left over hash lemma is more general and enables us to prove security of an extension discussed in Section 4.

The expanded system $\mathcal{E}_1$ As discussed in Section 3.1, a technical difficulty in the proof is that not every $(\ell+1)$-vector over $\mathbb{G}$ is a valid ciphertext in $\mathcal{E}$. We therefore introduce an "expanded version" of our scheme (denoted $\mathcal{E}_1$) that has the same secret key and decryption procedure, but a larger public key. In this system every vector in $\mathbb{G}^{1\times(\ell+1)}$ is a valid ciphertext. We later prove that $\mathcal{E}_1$ is n-way KDM-secure with respect to $\mathcal{C}_{n\ell}$, and then use it to deduce also the KDM-security of the original system $\mathcal{E}$.

- **Key Generation.** Let $\ell = \lceil 3\log_2 q \rceil$. Choose a random secret key $\boldsymbol{s} \xleftarrow{R} \{0,1\}^\ell \subset \mathbb{Z}_q^\ell$. Choose a random matrix $\mathit{\Psi} \xleftarrow{R} \mathrm{Rk}_\ell\left(\mathbb{G}^{(\ell+1)\times\ell}\right)$, and set $\mathit{\Phi} := \left(\mathit{\Psi} \middle| -\mathit{\Psi}\boldsymbol{s}\right) \in \mathbb{G}^{(\ell+1)\times(\ell+1)}$. Define the public and secret keys to be

$$\mathrm{pk} := \mathit{\Phi} \quad \text{and} \quad \mathrm{sk} := \boldsymbol{s}\gamma$$

  That is, the secret key is as in the system $\mathcal{E}$, but we use an expanded public key $\mathit{\Phi}$, which is a matrix of $\ell+1$ public keys from $\mathcal{E}$ (all with respect to the same secret key $\boldsymbol{s}$).
- **Encryption.** To encrypt an element $\mu \in \mathbb{G}$, choose a random row vector $\boldsymbol{r} \xleftarrow{R} \mathbb{Z}_q^{1\times(\ell+1)}$ and output the ciphertext

$$\xi \leftarrow \boldsymbol{r}\mathit{\Phi} + \left(0^{1\times\ell} \middle| \mu\right) \in \mathbb{G}^{1\times(\ell+1)}$$

  This is similar to the original system $\mathcal{E}$, except that instead of a random multiple of the public-key vector as in Eq. (2), here we use a random linear combination of all the rows of the expanded public key $\mathit{\Phi}$.
- **Decryption.** Decryption is the same as in $\mathcal{E}$. Decryption works since the decoded secret key $\left(\boldsymbol{s}^\top \middle| 1\right)^\top$ is orthogonal to all the rows of the expanded public key $\mathit{\Phi}$.

We stress that the main difference between $\mathcal{E}$ and $\mathcal{E}_1$ is that in $\mathcal{E}$ the public key is just one vector orthogonal to the decoded secret key. In $\mathcal{E}_1$, on the other hand, the expanded public key spans the entire (ℓ-dimensional) subspace orthogonal to the decoded secret key. Jumping ahead, we will later show that under DDH, the adversary cannot distinguish between ciphertext vectors in $\mathcal{E}$ (taken from a 1-dimensional subspace) and ciphertext vectors in $\mathcal{E}_1$ (taken from an ℓ-dimensional subspace). Thus essentially the only difference from the adversary's perspective is that in $\mathcal{E}_1$ it sees more vectors in the public key.
In the proof below we use the following simple facts about $\mathcal{E}_1$:

Totality and uniformity. For any secret key sk with public key $\mathit{\Phi}$ and any element μ, if a ciphertext $\boldsymbol{\xi}$ decrypts to μ using sk, then $\boldsymbol{\xi}$ is a possible output of $\mathsf{E}(\mathit{\Phi}, \mu)$, i.e. a valid encryption of μ. Furthermore, all possible outputs of $\mathsf{E}(\mathit{\Phi}, \mu)$ are equally likely.

Public-key blinding. Let $\mathit{\Phi} \in \mathbb{G}^{(\ell+1)\times(\ell+1)}$ be a public key for some secret key sk and let R be a random invertible matrix, $R \xleftarrow{R} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$. Then blind-pk$(\mathit{\Phi}) := R\mathit{\Phi}$ is a uniformly random public key for sk. Furthermore, encryption with $\mathit{\Phi}$ and with $R\mathit{\Phi}$ produce the same distribution of ciphertexts.

Ciphertext blinding. Let $\mathit{\Phi} \in \mathbb{G}^{(\ell+1)\times(\ell+1)}$ be a public key, and let $\boldsymbol{\xi}$ be any encryption of $\mu \in \mathbb{G}$ with respect to $\mathit{\Phi}$. Let $\boldsymbol{r} \xleftarrow{R} \mathbb{Z}_q^{1\times(\ell+1)}$ be a random row vector, then blind-ct$(\mathit{\Phi}, \boldsymbol{\xi}) := \boldsymbol{r}\,\mathit{\Phi} + \boldsymbol{\xi}$ draws uniformly at random from $\mathsf{E}(\mathit{\Phi}, \mu)$.

Total blinding. If instead of being a valid public key, $\varPhi$ is a matrix of full rank $\ell+1$, then the output of blind-ct$(\varPhi, \boldsymbol{\xi})$ is uniformly random in $\mathbb{G}^{1\times(\ell+1)}$.

Self-referential encryption. Let sk $= (\gamma_1, \ldots, \gamma_\ell)^\top \in \mathbb{G}^\ell$ be a secret key with public key $\varPhi$. Denoting by $e_i \in \{0,1\}^\ell$ the unit vector with 1 in position i and 0 elsewhere, we have that $\left(\gamma e_i \middle| 0\right)$ is an encryption of the secret-key element γ_i with respect to $\varPhi$.

Plaintext homomorphism. Let $f(\boldsymbol{\mu}) = \boldsymbol{a} \cdot \boldsymbol{\mu} + \beta$ be an affine function from $\mathbb{G}^n$ to $\mathbb{G}$. Fix some vector $\boldsymbol{\mu} \in \mathbb{G}^n$, let $\varPhi$ be a public key, and let $\varXi \in \mathbb{G}^{n\times(\ell+1)}$ be a matrix whose i'th row is an encryption of μ_i with respect to $\varPhi$. Then $\boldsymbol{a}\varXi + \left(0^{1\times\ell} \middle| \beta\right)$ is an encryption of $f(\boldsymbol{\mu})$ with respect to $\varPhi$.

Secret-key homomorphism. Let $\boldsymbol{s} \in \{0,1\}^\ell$ be used for a secret key with public key $\varPhi$, and let $\boldsymbol{\xi} \overset{R}{\leftarrow} \mathsf{E}(\varPhi, \mu)$ be an encryption of an element $\mu \in \mathbb{G}$. Let $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ be an invertible affine function from $\mathbb{Z}_q^\ell$ to $\mathbb{Z}_q^\ell$, and set

$$M_f := \left(\begin{array}{c|c} A & \boldsymbol{b} \\ \hline 0^{1\times\ell} & 1 \end{array}\right) \quad \text{so that} \quad M_f \left(\boldsymbol{x}^\top \middle| 1\right)^\top = \left(f(\boldsymbol{x})^\top \middle| 1\right)^\top$$

Suppose that $f(\boldsymbol{s}) \in \{0,1\}^\ell$ (so $f(\boldsymbol{s})$ can be used for a secret key). Then $\varPhi\, M_f^{-1}$ is a public key for $f(\boldsymbol{s})$, and $\boldsymbol{\xi} M_f^{-1}$ is an encryption of μ with public key $\varPhi\, M_f^{-1}$. In particular, extend the xor function $\oplus$ to $\mathbb{Z}_q \times \{0,1\} \to \mathbb{Z}_q$ by

$$x \oplus 0 := x \quad \text{and} \quad x \oplus 1 := 1 - x$$

and extend it to vectors by applying it element-wise. Then for a fixed $\boldsymbol{a} \in \{0,1\}^\ell$, the function $f(\boldsymbol{s}) := \boldsymbol{s} \oplus \boldsymbol{a}$ is an affine function, so we can compute a public key and ciphertext vectors for $\boldsymbol{s} \oplus \boldsymbol{a}$ from a public key and ciphertext vectors for $\boldsymbol{s}$.

$\mathcal{E}_1$ is KDM-secure with respect to $\mathcal{C}_{n\ell}$

Theorem 2. *For any $\mathcal{C}_{n\ell}$-KDM-adversary $\mathcal{A}$ against $\mathcal{E}_1$ there exists a DDH-adversary $\mathcal{B}$ (whose running time is about the same as that of $\mathcal{A}$) such that*

$$\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1] \leq (2\ell - 1)\,\mathrm{DDH\,Adv}[\mathcal{B}, \mathbb{G}] + 1/q$$

Proof. We present this proof as a series of games, and we let w_i denote the probability that the adversary wins Game i.

Game 0. This game is identical to the $\mathcal{C}_{n\ell}$-KDM-security game defined in Section 2.1. By definition,

$$\left| w_0 - \frac{1}{2} \right| = \mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1] \tag{3}$$

Game 1. Game 1 looks the same as Game 0 to the adversary, but the challenger does not use the secret keys internally. For setup:

- The challenger generates a secret key $\boldsymbol{s} \overset{R}{\leftarrow} \{0,1\}^\ell$ with public key Φ, and then "forgets" $\boldsymbol{s}$. That is, the challenger does not use $\boldsymbol{s}$ for the rest of Game 1.
- The challenger chooses n random vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \overset{R}{\leftarrow} \{0,1\}^\ell$. It then produces a view to the adversary that is consistent with the n secret keys $\mathrm{sk}_i := (\boldsymbol{s} \oplus \boldsymbol{a}_i)\gamma$, but without ever using the "forgotten" $\boldsymbol{s}$.
- For each $i \in [1, n]$, the challenger uses the **secret-key homomorphism** and **public-key blinding** properties of $\mathcal{E}_1$ to generate a uniformly random public key pk_i for sk_i from $(\Phi, \boldsymbol{a}_i)$.

For brevity, let $\boldsymbol{\sigma} := (\ \mathrm{sk}_1^\top \mid \mathrm{sk}_2^\top \mid \ \ldots \ \mid \mathrm{sk}_n^\top\ )^\top$ denote the concatenation of the encoded secret keys (but the challenger does not use the value of $\boldsymbol{\sigma}$). To compute $\mathsf{E}(\mathrm{pk}_i, f(\boldsymbol{\sigma}))$ for an affine function f in $\mathcal{C}_{n\ell}$:

- For each $j \in [1, n]$, the challenger uses the **self-referential encryption** property to generate an encryption $\mathsf{E}(\mathrm{pk}_j, \mu)$ for every element $\mu \in \mathrm{sk}_j$, and uses **secret-key homomorphism** to transform it into an encryption under pk_i, $\mathsf{E}(\mathrm{pk}_i, \mu)$.
- The challenger concatenates these to obtain a matrix Ξ of encryptions under pk_i of all the elements in $\boldsymbol{\sigma}$.
- The challenger uses the **plaintext homomorphism** property to generate an encryption $\boldsymbol{\xi} \leftarrow \mathsf{E}(\mathrm{pk}_i, f(\boldsymbol{\sigma}))$.
- The challenger sends $\text{blind-ct}(\mathrm{pk}_i, \boldsymbol{\xi})$ to the adversary.

The distribution of secret keys, public keys and ciphertexts is identical to Game 0, so

$$w_1 = w_0 \tag{4}$$

Informally, the challenger has used a single public key Φ to generate an entire clique of ciphertexts, without knowing any of their secret keys. It remains to show formally that this gives the adversary no useful information.

The remaining games will be identical to Game 1, except that the initial public key Φ will be computed differently.

Game 2. In Game 2, the challenger does:

$$\Psi \overset{R}{\leftarrow} \mathrm{Rk}_1\left(\mathbb{G}^{(\ell+1)\times\ell}\right) \quad \text{and} \quad \Phi \leftarrow \left(\Psi \middle| -\Psi\boldsymbol{s}\right) \in \mathbb{G}^{(\ell+1)\times(\ell+1)}$$

This is the same procedure used in Game 1, except that now Ψ has rank 1 instead of rank ℓ. Lemma 1 tells us that there is a DDH-adversary $\mathcal{B}$, running in about the same time as $\mathcal{A}$, such that

$$|w_2 - w_1| \le (\ell - 1)\,\mathrm{DDH\,Adv}[\mathcal{A}, \mathbb{G}] \tag{5}$$

Note that Ψ here may be computed by choosing random nonzero vectors $\boldsymbol{\psi} \overset{R}{\leftarrow} \mathbb{G}^{\ell+1}$ and $\boldsymbol{r} \overset{R}{\leftarrow} \mathbb{Z}_q^\ell$, and setting $\Psi \leftarrow \boldsymbol{\psi} \times \boldsymbol{r}$. Thus we see that $\Phi = \boldsymbol{\psi} \times \left(\boldsymbol{r}^\top \middle| -\boldsymbol{r}\cdot\boldsymbol{s}\right)^\top$ is $(1/q)$-uniform in $\mathrm{Rk}_1\left(\mathbb{G}^{(\ell+1)\times(\ell+1)}\right)$ by Corollary 1.

Game 3. Since $\varPhi$ is $(1/q)$-uniform in $\mathrm{Rk}_1\left(\mathbb{G}^{(\ell+1)\times(\ell+1)}\right)$, we can replace it by a random matrix in $\mathrm{Rk}_1\left(\mathbb{G}^{(\ell+1)\times(\ell+1)}\right)$. Thus, Game 3 is the same as Game 2, except that $\varPhi \xleftarrow{R} \mathrm{Rk}_1\left(\mathbb{G}^{(\ell+1)\times(\ell+1)}\right)$. Then

$$|w_3 - w_2| \leq 1/q \tag{6}$$

Note that in Game 3 the secret $\boldsymbol{s}$ is not used anywhere.

Game 4. Game 4 is the same as Game 3, except that $\varPhi \xleftarrow{R} \mathrm{Rk}_{\ell+1}\left(\mathbb{G}^{(\ell+1)\times(\ell+1)}\right)$. By the **total blinding** property of $\mathcal{E}_1$, the ciphertexts returned to the adversary are all uniformly random, regardless of the challenger's bit b. Therefore,

$$w_4 = \frac{1}{2} \tag{7}$$

On the other hand, by lemma 1, there exists a DDH-adversary $\mathcal{B}$, running in about the same time as $\mathcal{A}$, such that

$$|w_4 - w_3| \leq \ell\, \mathrm{DDH\,Adv}[\mathcal{B}, \mathbb{G}] \tag{8}$$

Combining equations (3) through (8), we find that

$$\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}_1] \leq (2\ell - 1)\, \mathrm{DDH\,Adv}[\mathcal{B}, \mathbb{G}] + 1/q$$

This completes the proof of Theorem 2. □

$\mathcal{E}$ is KDM-secure with respect to $\mathcal{C}_{n\ell}$ We now deduce the KDM-security of $\mathcal{E}$ from that of $\mathcal{E}_1$.

Lemma 3. *For any $\mathcal{C}_{n\ell}$-KDM adversary $\mathcal{A}$ against $\mathcal{E}$, there is a DDH-adversary $\mathcal{B}_1$ and a $\mathcal{C}_{n\ell}$-KDM adversary $\mathcal{B}_2$ against $\mathcal{E}_1$, both running in about the same time as $\mathcal{A}$, such that*

$$\mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}] \leq (\ell - 1) \cdot \mathrm{DDH\,Adv}[\mathcal{B}_1, \mathbb{G}] + \mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{B}_2, \mathcal{E}_1]$$

Proof. We present the proof as a series of games. Let w_i denote the probability that the adversary $\mathcal{A}$ wins Game i.

Game 0. Game 0 is identical to the $\mathcal{C}_{n\ell}$-KDM-security game with respect to $\mathcal{E}$ defined in Section 2.1. By definition,

$$\left| w_0 - \frac{1}{2} \right| = \mathrm{KDM}^{(\mathrm{n})}\mathrm{Adv}[\mathcal{A}, \mathcal{E}]$$

Game 1. Game 1 is the same as Game 0, except that the challenger generates public keys and encryptions in a different but equivalent way. Specifically,

- The challenger chooses a random rank-1 matrix $\varPsi_0 \xleftarrow{R} \mathrm{Rk}_1\left(G^{(\ell+1)\times\ell}\right)$.

- The challenger chooses n secret keys $\boldsymbol{s}_i \overset{R}{\leftarrow} \{0,1\}^\ell$, for $i = 1, \ldots, n$. It creates the corresponding n public keys as follows. For $i \in [1, n]$ generate the public key pk_i by choosing two random invertible matrices $L_i \overset{R}{\leftarrow} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$ and $R_i \overset{R}{\leftarrow} \mathrm{GL}_\ell(\mathbb{Z}_q)$ and setting

  $$\Psi_i \leftarrow L_i \, \Psi_0 \, R_i \qquad \text{and} \qquad \mathrm{pk}_i := \Phi_i \leftarrow \big(\Psi_i \big| {-\Psi_i \boldsymbol{s}_i}\big).$$

  Note that the matrix Ψ_i is a uniformly random rank-1 matrix and is independent of Ψ_0.
- For each $i \in [1, n]$, the challenger chooses a random nonzero row of the public key Φ_i, and sends it to the adversary as the $\mathcal{E}$-public-key φ_i. This row is nonzero, random and orthogonal to $\boldsymbol{s}_i$ by construction, so it is a valid public key for $\boldsymbol{s}_i$ under $\mathcal{E}$.
- When answering queries, instead of encrypting a message $\boldsymbol{\mu}$ with φ_i under the system $\mathcal{E}$, the challenger encrypts it under $\mathcal{E}_1$ using Φ_i as the public key. In other words, it responds with $R\,\Phi_i + \big(0 \big| \boldsymbol{\mu}\big)$ where $R \overset{R}{\leftarrow} \mathbb{Z}_q^{n\times(\ell+1)}$. Note that Φ_i is not a valid public key for $\mathcal{E}_1$, but only because it has rank 1 instead of rank ℓ.

Because Φ_i has rank 1, all rows of Φ_i are multiples of φ_i. Therefore, the distributions of ciphertexts $\boldsymbol{r} \times \varphi_i + \big(0 \big| \boldsymbol{\mu}\big)$ under $\mathcal{E}$ and $R\,\Phi_i + \big(0 \big| \boldsymbol{\mu}\big)$ in Game 1 are identical. The distributions of public and secret keys are also identical, so the attacker sees the same distribution of messages as in Game 0. As a result, $w_1 = w_0$.

Game 2. Game 2 is the same as Game 1, except that the challenger chooses $\Psi_0 \overset{R}{\leftarrow} \mathrm{Rk}_\ell\left(\mathbb{G}^{(\ell+1)\times\ell}\right)$ so that Φ is a random, valid public key under $\mathcal{E}_1$. This is the only difference between Games 1 and 2. By Lemma 1, there is a DDH adversary $\mathcal{B}_1$, running in about the same time as $\mathcal{A}$, such that

$$|w_2 - w_1| \le (\ell - 1)\, \mathrm{DDH\,Adv}[\mathcal{B}_1, \mathbb{G}]$$

At this point the attacker is attacking $\mathcal{E}_1$, with all but one row of the public keys hidden. Call this process $\mathcal{B}_2$; then

$$\left| w_2 - \frac{1}{2} \right| = \mathrm{KDM^{(n)}Adv}[\mathcal{B}_2, \mathcal{E}_1]$$

so that

$$\mathrm{KDM^{(n)}Adv}[\mathcal{A}, \mathcal{E}] \le (\ell - 1)\, \mathrm{DDH\,Adv}[\mathcal{B}_1, \mathbb{G}] + \mathrm{KDM^{(n)}Adv}[\mathcal{B}_2, \mathcal{E}_1]$$

as claimed. □

Theorem 1 now follows by combining Theorem 2 with Lemma 3. □

4 Extensions

Security under the linear assumption. The linear assumption, introduced in [5], is a weaker assumption than DDH. Weaker versions of the linear assumption were studied in [14,20]. The proof of Theorem 2 generalizes easily to use these weaker versions of the linear assumption. In particular, to use the r-linear assumption one need only change the value of ℓ to $\ell := \lceil (r+2) \log_2 q \rceil$. This hurts efficiency, but bases security on a weaker assumption. Note that the DDH assumption is identical to the 1-linear assumption.

Shrinking the ciphertext and secret keys. Ciphertexts and secret keys in our system contain $\ell := \lceil 3 \log_2 q \rceil$ elements in $\mathbb{G}$ where $q = |\mathbb{G}|$. This size of ℓ is chosen so that secret keys have sufficient entropy to make the distribution in Corollary 1 be $(1/q)$-uniform.

Recall that the secret key sk in our system is an encoding of a vector $\boldsymbol{s} \in \{0,1\}^\ell$, namely $\mathrm{sk}_i := g^{\boldsymbol{s}_i}$ for $i = 1, \ldots, \ell$. The vector $\boldsymbol{s}$ had to be binary for two reasons. First, during decryption we need to recover $\boldsymbol{s}$ from its encoding sk. Second, the proof of Theorem 2 relied on the fact that a vector $\boldsymbol{s} \in \{0,1\}^\ell$ can be mapped to a random vector in $\{0,1\}^\ell$ using an appropriate random affine map (i.e. by xoring with a known random vector in $\{0,1\}^\ell$, which is an affine map).

Let T be the set of ℓ-tuples that contains all $\ell!$ permutations of $(1, 2, \ldots, \ell)$. It is not hard to see that T satisfies the two properties mentioned above: (1) if we encode an ℓ-tuple in T by exponentiation as before then decoding can be done efficiently during decryption, and (2) an element $\boldsymbol{s} \in T$ can be mapped to a random element in T by the appropriate random affine transformation, namely a random permutation matrix. Hence, the proof of the main theorem (Theorem 1) will go through unchanged if algorithm G chooses $\boldsymbol{s}$ at random in the set T. Since the set T is larger than the set $\{0,1\}^\ell$ — the former is of size $\ell!$ while the latter is of size 2^ℓ — we can use a smaller value of ℓ and still satisfy the entropy bounds of Corollary 1. In particular, it suffices to choose

$$\ell = \left\lceil \frac{4.5 \log_2 q}{\log_2 \log_2 q} \right\rceil \quad \text{so that} \quad \ell! > q^3$$

This shrinks ciphertexts and secret keys by a factor of $O(\log \log q)$ over the original system.

5 One-way encryption that is not 2-circular secure

Beyond constructing encryption systems for which we can prove circular security, one may ask the more fundamental question of "what does it really take" to get circular security. For example, can we obtain circular-secure encryption from CPA-secure encryption? Recent work casts doubt on our ability to prove such implications using standard tools [11], but does not shed light on the deeper question of the truth of it. In fact, today we cannot even rule out the possibility that every CPA-secure system is also n-circular secure for all $n \geq 2$.

In this section we try to make some progress toward ruling out this possibility. Ideally, one would like to exhibit a CPA-secure system that is not (say) 2-circular secure. Unfortunately, we did not find a candidate system. Instead, we show a weaker example of a one-way encryption system that breaks completely once an n-cycle of encryptions is published (for any n).

One-way encryption is defined using the following game between a challenger and an adversary $\mathcal{A}$ (over a plaintext space M):

init. The challenger runs algorithm $\mathsf{G}(\lambda)$ to generate a public/secret key pair (pk, sk). The challenger then picks a random message $m \overset{R}{\leftarrow} \mathcal{M}$ and sends to the adversary the public key pk and the ciphertext $c \leftarrow \mathsf{E}(\mathrm{pk}, m)$.

response. The adversary $\mathcal{A}$ responds with its guess $m' \in \mathcal{M}$ for the decryption of c and wins the game if $m = m'$.

We define $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}(\lambda)$ to be the adversary's advantage in wining this game.

Definition 3. *We say that a public key encryption scheme $\mathcal{E}$ provides* one-way encryption *if* $\mathsf{Adv}[\mathcal{A},\mathcal{E}](\lambda)$ *is a negligible function in λ for any adversary $\mathcal{A}$ that runs in expected polynomial time in λ.*

Let $\mathcal{E} = (\mathsf{G},\mathsf{E},\mathsf{D})$ be a one-way secure system for message space M, and we assume that the secret keys are contained in M. Consider an encryption scheme $\bar{\mathcal{E}} = (\bar{\mathsf{G}},\bar{\mathsf{E}},\bar{\mathsf{D}})$ that operates on pairs of messages (i.e., has message space $M \times M$).

Key generation. Run G twice to generate two public/secret keys pairs $(\mathrm{pk}_1, \mathrm{sk}_1)$ and $(\mathrm{pk}_2, \mathrm{sk}_2)$. Output $\bar{\mathrm{pk}} := \mathrm{pk}_1$ as the public key and $\bar{\mathrm{sk}} := (\mathrm{sk}_1, \mathrm{sk}_2)$ as the secret key.

Encryption. An encryption of a message (m_1, m_2) under $\bar{\mathrm{pk}} = \mathrm{pk}_1$ is the pair $\big(m_1, \mathsf{E}_{\mathrm{pk}_1}(m_2)\big)$.

Decryption. Given a ciphertext (a, b) and secret key $\bar{\mathrm{sk}} = (\mathrm{sk}_1, \mathrm{sk}_2)$, output the pair $(a, \mathsf{D}_{\mathrm{sk}_1}(b))$.

Claim 3. *The system $\bar{\mathcal{E}}$ above is a one-way encryption system if $\mathcal{E}$ is. However, an attacker seeing an encryption cycle (of any size) can find all the secret keys involved.*

The proof is straightforward, and is omitted here. The "However" part follows since an adversary seeing an encryption of a secret key $(\mathrm{sk}_1, \mathrm{sk}_2)$ under any public key gets sk_1 in the clear, and therefore can decrypt any message encrypted under the public key corresponding to $(\mathrm{sk}_1, \mathrm{sk}_2)$.

Remark: In $\bar{\mathcal{E}}$ the first half of the plaintext is transmitted in the clear. One can partially hide this part too, as follows:

Assume that we have a one-way permutation f on secret-keys of $\mathcal{E}$, and moreover that f is defined and is one-way on the entire message space of $\mathcal{E}$. Further assume that from any secret key we can efficiently compute a corresponding public key (which we denote by writing $\mathrm{pk} = P(\mathrm{sk})$). Then define a system $\tilde{\mathcal{E}} = (\tilde{\mathsf{G}},\tilde{\mathsf{E}},\tilde{\mathsf{D}})$ as follows:

Key generation. Run G twice to generate two public/secret keys pairs $(\mathrm{pk}_1, \mathrm{sk}_1)$ and $(\mathrm{pk}_2, \mathrm{sk}_2)$. Output $\tilde{\mathrm{pk}} := P(f(\mathrm{sk}_1))$ as the public key and $\tilde{\mathrm{sk}} := (\mathrm{sk}_1, \mathrm{sk}_2)$ as the secret key.

Encryption. An encryption of (m_1, m_2) under $\tilde{\mathrm{pk}}$ is $\big(f(m_1), \mathsf{E}_{\tilde{\mathrm{pk}}}(m_1), \mathsf{E}_{\tilde{\mathrm{pk}}}(m_2)\big)$.

Decryption. Given a ciphertext (a, b, c) and secret key $\bar{\mathrm{sk}} = (\mathrm{sk}_1, \mathrm{sk}_2)$, compute $\mathrm{sk} = f(\mathrm{sk}_1)$ and output the pair $(\mathsf{D}_{\mathrm{sk}}(b), \mathsf{D}_{\mathrm{sk}}(c))$.

Again, proving a claim analogous to Claim 3 is straightforward.

6 Conclusions

We presented the first encryption system that can be proved to be n-circular secure under chosen-plaintext attack in the standard model. Security is based on the Decision Diffie-Hellman assumption and holds even if the adversary is given affine functions of the secret keys. In addition, we constructed in Section 5 a simple system that is weakly secure, but breaks completely once a key-cycle is published.

An important remaining problem is to obtain circular security against chosen ciphertext attacks. Other interesting problems are to improve the performance of our system, and to construct a semantically secure system that becomes insecure once an n-encryption cycle is published.

References

1. P. Adao, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *in Proceedings of the 10th European Symposium on Research in Computer Security - ESORICS'05*, volume 3679 of *LNCS*, pages 374–396, Springer, 2005.
2. M. Bellare and C. Namprempre. Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In *Proceedings of Asiacrypt '00*, volume 1976 of *LNCS*, pages 531–545, Springer, 2000.
3. M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient encryption. In *proceedings of Asiacrypt'00*, volume 1976 of *LNCS*, pages 317–330, Springer, 2000.
4. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *proceedings of Selected Areas in Cryptography (SAC'02)*, volume 2595 of *LNCS*, pages 62–75, Springer, 2002.
5. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proceedings of CRYPTO'04*, volume 3152 of *LNCS*, pages 41–55, Springer, 2004.
6. J. Camenisch and A. Lysyanskaya. An efficient system, for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of Eurocrypt'01*, volume 2045 of *LNCS*, pages 93–118, Springer, 2001.
7. R. Cramer and V. Shoup. A practical cryptosystem provably secure under chosen ciphertext attack. In *proceedings of CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25, Springer, 1998.
8. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM J. of Computing*, 30(2):391–437, 2000.
9. P. Erdös and R. Hall. Probabilistic methods in group theory II. *Houston Math Journal*, 2:173–180, 1976.
10. S. Goldwasser and S. Micali. Probabilistic encryption. *Jour. of Computer and System Science*, 28(2):270–299, 1984.
11. Iftach Haitner and Thomas Holenstein. On the (Im)Possibility of Key Dependent Encryption. Cryptology ePrint Archive, 2008. `http://eprint.iacr.org/2008/164`.
12. S. Halevi and H. Krawczyk. Security under key-dependent inputs. In *proceedings of the 14th ACM conference on computer and communications security (CCS)*, 2007. full version `http://eprint.iacr.org/2007/315`.
13. J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
14. D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *proceedings of CRYPTO'07*, volume 4622 of *LNCS*, pages 553–571, Springer, 2007.
15. D. Hofheinz and D. Unruh. Towards key-dependent message security in the standard mode. In *proceedings of Eurocrypt'08*, volume 4965 of *LNCS*, pages 108–126, Springer, 2008.
16. J. Katz and M. Yung. Unforgeable encryption and adaptively secure modes of operation. In *Fast Software Encryption, FSE'00*, volume 1978 of *LNCS*, pages 284–299, Springer, 2000.
17. H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In *Proceedings of CRYPTO'01*, volume 2139 of *LNCS*, Springer, 2001.
18. P. Laud and R. Corin. Sound computational interpretation of formal encryption with composed keys. In *proceedings of 6th International Conference on Information Security and Cryptology - ICISC'03*, volume 2971 of *LNCS*, pages 55–66, Springer, 2003.
19. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *in proceedings of CRYPTO'91*, volume 576 of *LNCS*, pages 433–444, Springer, 1992.
20. H. Shacham. A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants. Cryptology ePrint Archive, 2007. `http://eprint.iacr.org/2007/074`.
21. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2005.