Efficient Circuit-Size Independent Public Key Encryption with KDM Security

Tal Malkin[1], Isamu Teranishi[1,2], and Moti Yung[1,3]

[1] Columbia University
[2] NEC Japan
[3] Google Inc.
tal@cs.columbia.edu, teranisi@ah.jp.nec.com, moti@cs.columbia.edu

Abstract. *Key Dependent Message (KDM) secure* encryption is a new area which has attracted much research in recent years. Roughly speaking, a KDM secure scheme w.r.t. a function set $\mathcal{F}$ provides security even if one encrypts a key dependent message $f(sk)$ for any $f \in \mathcal{F}$. We present a construction of an *efficient* public key encryption scheme which is KDM secure with respect to a large function set $\mathcal{F}$. Our function set is a function computable by a polynomial-size *Modular Arithmetic Circuit (MAC)*; we represent the set as *Straight Line Programs* computing multi-variable polynomials (an extended scheme includes all rational functions whose denominator and numerator are functions as above). Unlike previous schemes, our scheme is what we call *flexible*: the size of the ciphertext depends on the degree bound for the polynomials, and beyond this all parameters of the scheme are *completely independent* of the size of the function or the number of secret keys (users). We note that although KDM security has practical applications, all previous works in the standard model are either inefficient feasibility results when dealing with general circuits function sets, or are for a small set of functions such as linear functions. Efficiency of our scheme is dramatically improved compared to the previous feasibility results.

1 Introduction

The design of public key systems that are secure against attackers who are allowed to request ciphertexts that are a function of the system's secret keys is a very active area of research. The initial schemes designed in this area were called "circular" [CL01] and allowed encryption of a secret key or a linear function of a secret key; later, more general functions were considered and the security of these schemes was called *Key Dependent Message (KDM) security* [BRS02]. In particular, we say that a public-key encryption (PKE) scheme is *KDM*[$\mathcal{F}$] *secure* (where $\mathcal{F}$ is a class of function), if it is secure even against an adversary who is given public keys $pk_1, \ldots, pk_n$ and has access to encryption of key dependent messages $f(sk_1, \ldots, sk_n)$ for adaptively selected functions $f \in \mathcal{F}$.

Originally motivated by the fact that in some systems keys encrypt other keys (by design or by misuse of protocols), recent research has revealed other important motivations for studying KDM security. On the theoretical side, KDM security can be used to "reconcile" the two fundamental views of security, indistinguishability based security and Dolev-Yao security [AR00,BRS02,ABHS05, BPS08]. This notion also has surprising connection with other fundamental notions, cryptographic agility [ABBC10], and obfuscation [CKVW10]. On the practical side, KDM security is crucial for designing some recent cryptographic protocols. For instance, this notion is used in an anonymous credential system [CL01], where a KDM secure encryption is used to discourage delegation of credentials. Another example is fully homomorphic encryption, where KDM security is used to achieve the full unbounded construction of [G09].

Almost all previous work on KDM security focused on finding KDM[$\mathcal{F}$] (standard model) secure public key encryption schemes for a function class $\mathcal{F}$ that is as large as possible, without much consideration to efficiency. KDM security for the largest set of functions – all functions of bounded Boolean circuit size – was achieved by Barak, Haitner, Hofheinz and Ishai [BHHI10], following previous works such as [BHHO08,BGK09]. However, the schemes in all these works are quite inefficient. For instance, even the most efficient seminal scheme of [BHHO08] requires us to compute $\Theta(\kappa)$ exponentiations over the underlying group $\mathbb{G}$ per each bit of secret key. Here κ is a security parameter. This incurs a factor $\Theta(\kappa^2)$ loss over the standard ElGamal encryption where one can encrypt κ bit by executing $\Theta(1)$ exponentiations. The work of Applebaum, Cash, Peikert, and Sahai [ACPS09] is the only one which explored efficient KDM secure schemes and proposed a much more efficient scheme than others. However, that work is KDM-secure only for linear classes of functions. We discuss previous work in more detail in Section 1.7.

1.1 Our Goals

Efficient Encryption with a Large Set of Queries. In this work we consider the challenge of designing an *efficient* KDM secure scheme that allows the attacker a *large* set of functions $\mathcal{F}$ over the secret key to draw from. The efficiency of the scheme should be comparable with that of the ElGamal encryption (at least for a constant size function family) which is a block-wise semantically secure encryption (and dramatically better than previous KDM-secure works [BHHO08,BGK09,BHHI10] that pay a factor of at least $\Theta(\kappa^2)$ over ElGamal).

Constructing efficient KDM[$\mathcal{F}$] secure scheme for large $\mathcal{F}$ is challenging, and the techniques of previous works seem insufficient. Indeed, all previous works in the standard model are either inefficient feasibility results or possess some noticeable overhead [BHHO08,CCS09, BHHI10,BGK09,BG10], or for a small set of functions such as linear functions [ACPS09].

Flexible Parameters. Another important factor which was ignored in past investigations, is what we call *flexibility parameters* of a scheme, dealing with restrictions on the choice of parameters of functions in $\mathcal{F}$, and when those need to be bounded (determined). For example, consider the number n of keys (users) in $f(sk_1, \ldots, sk_n) \in \mathcal{F}$. Some schemes (e.g., [BG10]) do not allow to select n

flexibly, but rather require the maximum n to be fixed before key generation, and KDM security proof is subject to the scheme being so restricted. Clearly, a flexible scheme that allows unbounded (freely determined) n even after key generation is desirable.

For flexibility determination we will consider the following parameters: *number of keys*, the *size of the circuit* of the function, and its *degree as a polynomial.* For these parameters we will ask whether they need to be determined (bounded) as an input (1) at key generation, (2) at encryption time, or (3) remain unbounded throughout. These are listed in order of increased flexibility, and the more flexible a KDM-scheme is, the more desirable.

1.2 Our Results

We design block-wise encryption (i.e., a scheme encrypting messages as blocks rather than bit by bit) that is efficient, flexible, and provides KDM-security against a large set of functions, based on the DCR assumption.

Roughly, our scheme provides KDM security for polynomials and rational functions (ratios of two polynomials) over a ring of integers (modulo N), with the flexibility of allowing unbounded number of keys, circuit of unbounded size, computing a polynomial whose degree is bounded only at encryption time. This is the first time flexibility is defined, and the first time that KDM security has been achieved with this level of flexibility (no dependence on number of keys, no dependence at key generation time on the circuit, and depending only on the degree, but not the size, of the circuit at encryption time.) We also give a general *triple mode proof framework* for proving KDM security, which was implicitly used in previous works including ours.

We elaborate on these contributions below.

1.3 Function Classes

A function f is called *MAC (Modular Arithmetic Circuit)* if there exists a polynomial-size circuit for f whose inputs are variables $X_1, \ldots, X_n$ and constants of $\mathbb{Z}_K$ and whose gates are $+$, $-$, or $\cdot$ over a ring $\mathbb{Z}_K$. That is, f is MAC computable iff it can be computed from variables and constants of $\mathbb{Z}_K$ by applying $+$, $-$, and $\cdot$ modulo K a polynomial number of times (this is also referred to as a *straight line program* with n variables over $\mathbb{Z}_K$ which can be viewed as computing a polynomial function).

The set of functions $\mathcal{MAC}_d[K]$ contains functions whose total degree (as a polynomial) is not more than $d = \mathsf{poly}(\kappa)$, and which are MAC computable over $\mathbb{Z}_K$. The set of functions $\mathcal{Q}(\mathcal{MAC}_d[K])$ is the set of functions which can be represented by a ratio (division) of two MAC computable functions. These two are the sets of functions that our two schemes (for different K) will be KDM-secure against, respectively.

Richness of Function Classes. These classes are quite large. To start with, $\mathcal{MAC}_d[K]$ includes all functions which are represented by polynomial length formula with total degree $\leq d$, where a *formula* is a (well-formed) word on the set of alphabets $\{X_1, \ldots, X_n, +, \cdot, ^m, (,), a \mid m \in \mathbb{Z}, a \in \mathbb{Z}_{N^{s-1}}\}$. $\mathcal{Q}(\mathcal{MAC}_d[K])$ includes all functions that are represented by such a formula that also allows division (or inverse), such as

$$((2X_1+X_2+\cdots+X_n)^{10}+(X_1+4)\cdots(X_n+4)(X_2+4X_3)^{-1})^2+3({X_3}^{-3}-2{X_2}^2X_1)^2 \bmod K.$$

In fact, the $\mathcal{MAC}_d[K]$ class is much richer, as such formulas can be re-interpreted as log depth circuits, while MACs can have polynomial depth. MACs can be simulated by a straight line program with as many variables as the depth of the circuit (i.e., simply traverse the circuit in topological order); the result is a polynomial and we only require its degree to be bounded (at most d). Note that $\mathcal{MAC}_d[K]$ can contain polynomials that have exponentially many terms. The simplest example of such a function is

$$f(X_1, \ldots, X_n) = (X_1 + \cdots + X_n)^d \bmod K \qquad \text{for } n = \mathsf{poly}(\kappa), d = \mathsf{poly}(\kappa)$$

This function can clearly be computed by a MAC (with polynomial number of gates), but it has an exponential number of terms $\{{X_1}^{\varepsilon_1} \cdots {X_n}^{\varepsilon_n} \mid \varepsilon_1 + \cdots + \varepsilon_n = d\}$ when expanded.

On the other hand, by definition, these classes cannot compute functions that have an exponential degree (as we need the polynomial degree for our KDM secure construction). For example, $\mathcal{MAC}_d[K]$ does not contain $f(X) = X^{2^\kappa}$ (even though this f can be computed by a polynomial size MAC).

1.4 Properties of Proposed Schemes

We construct two efficient block-wise KDM secure PKE schemes as following:

KDM Security: Our schemes are KDM$[\mathcal{MAC}_d[N^{s-1}]]$ and KDM$[\mathcal{Q}(\mathcal{MAC}_d[N])]$ secure respectively, where N is the product of two safe primes and $s \geq 2$.

Computational Costs: An encryption and a decryption of our first scheme require only $2d+4$ $(d+2)$ exponentiations in $\mathbb{Z}_{N^s}$ when one encrypts (decrypts) a ciphertext of $f(sk_1, \ldots, sk_n)$ with degree d. (The costs double for our second scheme).

Flexibilities of Parameters: In our schemes, for the first time, the number n of keys of a function $f(sk_1, \ldots, sk_n)$, the number ℓ of $\{+, -, \cdot\}$ in a MAC computing f (i.e., circuit size), and the total degree d of f can be selected flexibly by an adversary. Specifically, our schemes are KDM secure even under the condition that an adversary can choose these parameters arbitrarily and adaptively, where n, ℓ are completely unrestricted, and d is needed as input at the encryption stage. (The obvious upper bound for these parameters is the number of steps of the adversary herself, which is some polynomial in κ.) This also means that the party who is encrypting can choose which d to protect against for each encryption, depending on the level of sensitivity or perceived KDM-attack risk for that encryption.

Moreover, the efficiency of our schemes (both in terms of computational cost and ciphertext length) do not depend on n and ℓ. Our schemes therefore remain efficient even if these parameters are quite big. This is in contrast with recent schemes, as will be compared below.

1.5 Triple Mode Proof Framework

For our security proofs, we give a general framework for proving KDM security, the *triple mode proof framework*, which clarifies the structure of the proof, and highlights the crucial parts. Intuitively, the definition of KDM security involves an adversary that can access an encryption oracle, asking for encryptions of $f(sk_1, \ldots, sk_n)$, and getting either the correct key-dependent encryptions, or random bit encryptions; the adversary should not be able to distinguish between these two cases. To prove security, we need to construct a simulator which can use any such distinguishing adversary to break the underlying assumption. However, this is problematic, because without the secret key, it is not clear how the simulator can compute encryptions of key-dependent functions, however, with the secret key, these two cases could be distinguishable, and breaking the underlying assumption is no contradiction.

Our triple mode proof framework solves this issue by preparing three games (or "modes") for the security proof, and using *two* simulators, where the first one knows the secret key but the second one does not. The first and last game correspond to the usual key-dependent vs. random encryption, as above. The key idea is to find an intermediate game where the encryptions are independent of the secret key, yet it is indistinguishable from the first game *even given the secret key.*

The suggested notion unifies security techniques, since it can be shown that known standard model KDM secure schemes [BHHO08, ACPS09,BG10] as well as ours have the structure suitable for the triple mode proof frameworks.

1.6 Techniques

Here we describe our main ideas in a way that is informal and inaccurate, yet hopefully it provides good intuition. We use the following approach.

- Construct an efficient block-wise KDM secure scheme $\mathcal{PKE}$ w.r.t. moderately large and simple set $\mathcal{F}$.
- Reduce the KDM security of $\mathcal{PKE}$ w.r.t. the quite large and complex set $\mathcal{MAC}_d[N^{s-1}]$ into KDM security of it w.r.t. $\mathcal{F}$, by "compressing" the complex structure of MAC into the simple structure of $\mathcal{F}$.

It is important to choose $\mathcal{F}$ carefully, so as it is not too large or too small. We choose $\mathcal{F}$ to be the set of univariate polynomials $f(X) = \sum_{j=0}^{d} a_j X^j$, and construct a KDM scheme w.r.t. $\mathcal{F}$ based on new idea, the *cascaded Paillier ElGamal* and show it satisfies KDM security.

KDM scheme for uni-variate polynomials : cascaded Paillier ElGamal. Our starting point is previous work (in particular [BG10]), which achieved KDM-security for linear functions on bits. Transforming that to block-wise linear functions on entire secret keys is straight forward.[4]

An encryption of a message M in the resulting [BG10] scheme is ciphertext of the form $(C_1, A(M)C_2)$, where $A(M)$ is the only part that depends on the message (a la ElGamal encryption); concretely our starting point was [KTY09]. The KDM-security relies on the fact that when the message is a linear function $f(x) = ax + b$ of the secret key x, its encryption $(C_1, A(ax + b)C_2)$ is indistinguishable from the encryption $(A(a)C_1, A(b)C_2)$, which now no longer depends on $(ax + b)$, but only on a, b (and thus can be simulated using the secrecy of the key x).

To extend this to a function in $\mathcal{F}$ that is a degree d polynomial $f(x)$, we write $f(x) = f'(x)x + b$, and can say, similarly to above, that the ciphertext $(C_1, A(f'(x)x + b)C_2)$ is indistinguishable from $(A(f'(x))C_1, A(b)C_2)$. Now the right term is independent of the secret key, and the left term does depend on the secret key, but only as a degree $d - 1$ polynomial $f'(x)$.

Our "cascaded Paillier ElGamal" scheme thus ElGamal encrypts the left element to get $(C_1', A(f'(x))C_2')$, and apply the same idea recursively to reduce the degree of $f'(x)$ by one. We may continue recursively constructing these pairs, each time encrypting the left element with a fresh encryption. The final ciphertext we output is a tuple consisting of all right elements, and the last left element.

Reduction from MACs to Univariate Polynomials. In order to achieve our general class, for many keys, we reduce their number by setting secret keys sk_i to the sum $\mu+\alpha_i$ of one "secret" μ and a "difference" α_i from μ.[5] Then a multivariate polynomial $f(sk_1, \ldots, sk_n)$ computed by an MAC can be re-interpret as a univariate polynomial $\phi(\mu) = f(\mu + \alpha_1, \ldots, \mu + \alpha_n)$ of μ. Namely, KDM security w.r.t. $f(sk_1, \ldots, sk_n)$ is reduced to KDM security w.r.t. a univariate polynomial $\phi(\mu)$.

The crucial point of the above reduction is that the coefficients of $\phi(\mu)$ can be computed in polynomial time if f is an element of $\mathcal{MAC}_d(N^{s-1})$. This fact enables simulators to remain polynomial time algorithms. This is why we use $\mathcal{MAC}_d(N^{s-1})$.

The Second Scheme: Security for Quotient of MACs. A ciphertext of our second scheme for a message M is a tuple $(C', C'') = (\mathsf{Enc}(MR), \mathsf{Enc}(R))$, where Enc is the encryption of the first scheme and R is a randomly selected element.

[4] [BG10] did not mention this, probably because they also consider other goals such as leakage resilience, for which they focused on bit functions.

[5] We note this idea has been used in several previous works to handle multiple keys, but in our case it provides much more powerful results for the class of functions, since we start from a polynomial degree function, rather than a constant degree one.

Intuitive meanings of C' and C'' are the encryptions of "numerator" and the "denominator" of a key dependent message $f(\overrightarrow{sk}) = f'(\overrightarrow{sk})/f''(\overrightarrow{sk})$ computed by two MAC computable functions f' and f''. Here $\overrightarrow{sk} = (sk_1, \ldots, sk_n)$.

Clearly, encryption (C', C'') of key dependent message $f(\overrightarrow{sk})$ has the same distribution as $(\mathsf{Enc}(f'(\overrightarrow{sk})S), \mathsf{Enc}(f'(\overrightarrow{sk})S))$ for randomly selected S. We therefore succeed in reducing the KDM security of the second scheme to KDM security of the encryptions $\mathsf{Enc}(f'(\overrightarrow{sk})S)$ and $\mathsf{Enc}(f'(\overrightarrow{sk})S)$ of the first scheme.

The only problem of the above idea is that the denominator $f''(\overrightarrow{sk})$ can be 0 and therefore $f = f'/f''$ cannot be defined in this case. But we can overcome this problem by modifying the scheme slightly and proving the security carefully.

1.7 Related Work and Comparison to Our Schemes

Fig.1 shows the comparison among the previous schemes and ours. Here κ is the security parameter. Note that all schemes except for [CCS09] are KDM-CPA secure, while [CCS09] is KDM-CCA2 secure.

Explanation of Fig.1: The "size" ℓ represents the number of gates in a MAC (resp. in a circuit) computing a function $f(sk_1, \ldots, sk_n)$ in the case of our schemes (resp. [BHHI10]). The column "Flexibility of Param." describes the flexibilities of the parameters n, d, and ℓ of a function $f(sk_1, \ldots, sk_n)$. "Key-Gen bounded" means that one has to fix the maximum of the parameter before the key generation, KDM security holds only when the parameter is less than the maximum, and efficiency of the scheme depends on this maximum. "Enc bounded" means that we do not have to fix such maximums, and KDM security hold for all values of the parameter, but the parameter is needed for encryption, and efficiency of the scheme depends on its value. "Unbounded" means the scheme (at all stages) is independent from the value of this parameter.

The column "|Ciphertext| per |Message|" represents the ratio between the ciphertext length and the message length.

We note that we can improve properties of known schemes in Fig.1 using known techniques:

- Using the technique of [ACPS09], "|Ciphertext|/|Message|" of [BHHO08] and [BG10] can be reduced to $O(1)$.
- If one restricts the function to polynomials, [BHHI10] can be unbounded in n.

Comparison with [ACPS09]: They deal with linear functions only compared to our larger set of MACs, and they are based on a lattice-based assumption, LWE, while we employ the DCR assumption.

Comparison with [BHHI10,A11]: These schemes achieve KDM secure schemes w.r.t. the largest set of functions (strictly richer than ours), though their schemes are merely a feasibility result, relying on and are application of the inefficient general secure computation. While it is important to know the feasibility of such KDM secure schemes, they are not comparable to schemes that are more efficient

	Functions	|Ciphertext| per |Message|	# of Users n	Flexibility of Parameters		
				max deg d	Size ℓ	Assum-ption
[BHHO08] [CCS09]	Linear of bits	$O(\kappa)$	Un-bounded	-	-	DDH
[BGK09]	Polynomial of Bits with deg= $O(1)$	$O(\kappa^{d+1})$		KeyGen	-	DDH LWE
[BG10] +[BGK09]		$O(n\kappa + \kappa^{d+1})$	KeyGen			QR DCR
[BHHI10] [A11]	Bounded Size Circuit	$O(n\mathsf{poly}(\kappa) + \kappa\ell)$	Enc	-	Enc	DDH LWE QR
[ACPS09]	Linear of block	$O(1)$	Un-bounded	-	-	LWE
First Scheme	$\mathcal{MAC}_d[N^{s-1}]$	$O(d)$		Enc	Un-bounded	DCR
Second Scheme	$\mathcal{Q}(\mathcal{MAC}_d[N])$					

Fig. 1. Parameters of Our Scheme and Previous Work

then including the encryption of the circuit, or ,as in our scheme, independent of the circuit size. This is especially true given the applications of such encryption schemes. The size ℓ of circuit is encryption bounded in [BHHI10,A11], while it is unbounded in our scheme, which requires only the degree d to be encryption bounded.

Comparison with [BHHO08,BGK09,BG10]: The first to achieve KDM security without a random oracle were [BHHO08]. Their scheme was used as a basis for [BGK09] and [BG10], who achieve KDM secure schemes w.r.t. the set of *constant-degree polynomials* of *bits* of secret keys (as we said they can describe also blocks of keys rather than bits). The degrees of polynomials in [BGK09,BG10] have to be bounded by small constant (because the ciphertext lengths in these schemes grow exponentially with this degree), and therefore are KeyGen bounded. In contrast, in our schemes the degree of the polynomials can be polynomial and is encryption bounded, and the number of terms can be super polynomial. For the scheme of [BG10] the number of users is KeyGen bounded, while it is unbounded in our schemes. Finally, the schemes [BGK09,BG10] (and to a lesser extent [BHHO08]) are quite less efficient than ours.

Other Related Works. The notion of KDM security was defined by Black, Rogaway, and Shrimpton [BRS02], although Camenisch and Lysyanskaya [CL01] independently defined a similar notion called *circular encryption* earlier. Earlier works of KDM security were studied in the random oracle model. [BDU08] showed that the well-known OAEP encryption is KDM secure. [HK07] generalize the notion of KDM to pseudorandom functions.

Constructing KDM secure schemes in the standard model was a long-standing open problem. It was partially solved by [HU08] for the case of a symmetric key encryption. The first PKE which is KDM secure in the standard model was proposed in the seminal work of Boneh, Halevi, Hamburg, and Ostrovsky [BHHO08]. The first CCA2 and KDM secure PKE was proposed by [CCS09]. A general transformation starting from KDM secure PKE for a certain class of functions and boosting it to a larger class was shown by [A11]. Examples of PKE which satisfy semantic security but not KDM (specifically, 2-circular) security were shown independently by [GH10] and [ABBC10].

[HH09] showed that KDM[$\mathcal{F}$] security of an encryption scheme for "quite large" $\mathcal{F}$ cannot be proved as long as the reduction's proof of security treats the function $f \in \mathcal{F}$ and the adversary as black boxes.

The connection between the adaptive Dolev-Yao model and generalized versions of KDM security are studied by [BPS08], while further connections of KDM security with agility and obfuscation are shown by [ABBC10] and [CKVW10], respectively.

We refer the reader to [MTY11] for a survey on KDM security results and applications.

2 Preliminaries

Notations and Terminologies: For a natural number n and $m \leq n$, let $[n]$ and $[m..n]$ be the sets $\{1, \dots, n\}$ and $\{m, \dots, n\}$ respectively. For a real number x, $\lfloor x \rfloor$ denote the largest integer not greater than x.

Polynomials and Rational Functions: For a polynomial $f(X_1, \dots, X_n) = \sum_{j_1,\dots,j_n} a_{j_1,\dots,j_n} {X_1}^{j_1} \cdots {X_1}^{j_n} \mod K$, the *(total) degree* $\deg f$ of f is $\max\{\sum_k j_k \mid a_{j_1,\dots,j_n} \neq 0 \bmod K\}$. A *rational function* over $\mathbb{Z}_K$ is a function which can be written as $f(X_1, \dots, X_n)/g(X_1, \dots, X_n)$ using two polynomials f and g over $\mathbb{Z}_K$.

Paillier Group: Let N be the product of two safe primes and T be $1+N$. Define three subsets of $\mathbb{Z}_{N^s}$, sets of Quadratic Residue, Square Composite Residuosity, and Root of the Unity, as follows:

- $\mathcal{QR}[N^s] = \{u^2 \bmod N^s \mid u \in \mathbb{Z}_{N^s}\}$
- $\mathcal{SCR}[N^s] = \{r^{2N} \bmod N^s \mid r \in \mathcal{QR}[N^s]\}$,
- $\mathcal{RU}[N^s] = \{T^M \bmod N^s \mid M \in [0..N^{s-1}]\}$.

Theorem 1 ([P99,DJ01,KTY09]). *There exists a polynomial time computable bijective homomorphism $L : \mathcal{RU}[N^s] \rightarrow \mathbb{Z}_N$ satisfying the following property:*

$$\forall M \in \mathbb{Z}_{N^{s-1}} \quad : \quad L(T^M) = M \bmod N^{s-1}.$$

Moreover, the following property holds:

$$\mathcal{QR}[N^s] = \mathcal{SCR}[N^s] \times \mathcal{RU}[N^s].$$

Definition 2 (Decision Composite Residuosity (DCR) Assumption[P99,DJ01]) Let $s \geq 2$ be an integer. There exists a generator Gen of the product N of two safe primes such that the following value is negligible for κ for any polynomial time adversary A:

$$\begin{aligned}&|\Pr[N \leftarrow \mathsf{Gen}(1^\kappa), g \leftarrow \mathcal{SCR}[N^s], b \leftarrow \mathsf{A}(s, N, g) : b = 1]\\ &- \Pr[N \leftarrow \mathsf{Gen}(1^\kappa), g \leftarrow \mathcal{QR}[N^s], b \leftarrow \mathsf{A}(s, N, g) : b = 1]|.\end{aligned}$$

Our DCR assumption is subtly different from the original one [P99,DJ01], where g is taken from $\{r^N \bmod N^2 \mid r \in \mathbb{Z}_{N^s}\}$ (or $\mathbb{Z}_{N^s}$) in the first (or second) game, but ours clearly follows form the original one by squaring g.

2.1 KDM Security

Public Key Encryption Scheme: In this work, a public key encryption scheme $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ has generator Setup of a system parameter *prm*, such as a group description, and all users commonly use this parameter as inputs of the other three algorithms.

Description of Functions: As in previous works, we implicitly assume that each function f has some polynomial size *description* D. (In the case of our schemes, D is an MAC or MACs computing f.) We let f_D denote the function corresponding to D.

KDM Security: For a public key encryption scheme $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ and its secret key space SkSp and message space MeSp, let

$$\mathcal{F}^{(n)} \subset \{f \ : \ \mathsf{SkSp}^n \to \mathsf{MeSp}\}, \qquad \mathcal{F} = \cup_{n=1}^{\infty} \mathcal{F}^{(n)}.$$

To simplify, we assume that SkSp and MeSp depend only on the system parameter *prm*. For a natural number n and a bit b, consider the following game:

- $\mathsf{GameKDM}^b_{\mathsf{A}}[\mathcal{F}, n]$:

$$prm \leftarrow \mathsf{Setup}(1^\kappa), b' \leftarrow \mathsf{A}^{\mathcal{O}_{\mathsf{Kg}}, \mathcal{O}^{(b)}_{\mathsf{Enc}}}(prm, (pk_j)_{j \in [n]}), \text{ Output } b'.$$

Above, A is allowed to make polynomial number of queries adaptively:

- If A makes the i-th query new to $\mathcal{O}_{\mathsf{Kg}}$, it generates the i-th key pairs $(pk_i, sk_i) \leftarrow \mathsf{Kg}(prm)$ and sends pk_i as an answer.
- If A makes the i-th query (i, D) to $\mathcal{O}^{(b)}_{\mathsf{Enc}}$ where $i \in [n]$ and D is a description of a function of $\mathcal{F}^{(n)}$, the oracle answers the following C (below, $\mathfrak{o}$ be some fixed element of MeSp and n is the number of keys generated by $\mathcal{O}_{\mathsf{Kg}}$):

$$C \leftarrow \begin{cases} \mathsf{Enc}_{prm}(pk_i, f_D(sk_1, \ldots, sk_n)) & \text{if } b = 1 \\ \mathsf{Enc}_{prm}(pk_i, \mathfrak{o}) & \text{Otherwise.} \end{cases}$$

We say that $\mathcal{PKE}$ is *KDM*$[\mathcal{F}]$ *secure* if the following advantage is negligible for any n and any polynomial time algorithm A.

$$\mathsf{Adv.KDM}_{\mathsf{A}}[\mathcal{F}] = |\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]| \, .$$

One can easily check that the above definition is independent of the choice of $\mathfrak{o}$, if $\mathcal{PKE}$ satisfies indistinguishability.

Our definition of KDM security is stronger than that of the previous one [BRS02,BHHO08]: Ours allows an adversary to get new key adaptively while the previous one [BRS02,BHHO08] does not. (In other words, using the terminology of Section 1.4, the number n of keys becomes "unbounded.") Some known schemes (e.g., [BG10]) require the maximum n to be fixed before key generation, and KDM security can be proved only when n is less than the predetermined maximum. Our scheme does not require n to be fixed and therefore can be proved under our stronger KDM security definition.

2.2 Modular Arithmetic Circuit

Definition 3 (Modular Arithmetic Circuit (MAC)) A *Modular Arithmetic Circuit* D (MAC) is a circuit whose inputs are variables $X_1, \ldots, X_n$ and constants of $\mathbb{Z}_K$ and whose gates are $+$, $-$, or $\cdot$ over $\mathbb{Z}_K$. (We stress that the fan-out of each gate is unbounded.) For MAC D, the number of gates in D is called *size* of D.

Note that in our case MAC is equivalent to *straight line program* with unlimited number of registers. Clearly, a function computed by MAC is a polynomial over $\mathbb{Z}_K$. We let f_D denote f when MAC D computes f.

For natural numbers n, d, and ℓ, $\mathcal{MAC}_{n,d,\ell}[K]$ is the set of all rational functions which can be computed using some MAC D with size $\leq \ell$ and $\deg f_D \leq d$.[6] Here n indicates the number of inputs of D.

3 KDM Secure Scheme w.r.t Bounded Degree MAC

Cascaded Paillier ElGamal: Our scheme, called *d-cascaded Paillier ElGamal*, is computed recursively as follows. First, a "Paillier ElGamal" encryption $(e_0, c_0) = ({u_0}^{-1}, T^M v_0) \mod N^s$ of a message M is computed, where $T = 1+N$ and $(u_0, v_0) \leftarrow (g^{r_0}, h^{r_0})$. Next, the left component e_i of the ciphertext is encrypted by "Paillier ElGamal" encryption and $(e_{i+1}, c_{i+1}) = ({u_{i+1}}^{-1}, e_i v_{i+1})$ is obtained for $i = 1, \ldots, d-1$, where $(u_{i+1}, v_{i+1}) \leftarrow (g^{r_{i+1}}, h^{r_{i+1}})$. We finally let c_{d+1} be e_d. (Note that much of the encryption is not message dependent and can be performed off-line given the degree bound expected, as in ElGamal, but with much more performance gain).

The d-cascaded Paillier ElGamal encryption of message M is the tuple

$$C = (c_{d+1}, c_d, c_{d-1}, \ldots, c_0) = ({u_d}^{-1}, {u_{d-1}}^{-1} v_d, {u_{d-2}}^{-1} v_{d-1}, \ldots, T^M v_0).$$

Detailed Scheme: The detail of our scheme is as follows. Bellow, κ and ξ are security parameters and $s \geq 2$ and d are positive integers.

[6] The total degrees of the polynomials may not be computable from D in polynomial time. But this fact does not become a problem in our case, because we can easily compute an upper bound of the total degrees from D.

- $\mathsf{Setup}(1^\kappa)$: Generate the product N of two safe primes with $\lfloor \kappa/2 \rfloor$ bit lengths. Select $g \overset{\$}{\leftarrow} \mathcal{SCR}[N^s]$ randomly, and output $prm \leftarrow (s, N, g)$. (We will let T denote $1+N$.)
- $\mathsf{Kg}(prm)$: Select $sk \leftarrow x \overset{\$}{\leftarrow} [2^\xi \cdot \lfloor N/4 \rfloor]$ randomly, compute $pk \leftarrow h \leftarrow g^x \bmod N^s$, and output (pk, sk).
- $\mathsf{Enc}_{prm}(pk, M)$ for $M \in \mathbb{Z}_{N^{s-1}}$: Select $r_0, \ldots, r_d \overset{\$}{\leftarrow} [\lfloor N/4 \rfloor]$ randomly and output $C \leftarrow (c_{d+1}, \ldots, c_0)$, where

$$c_j \leftarrow \begin{cases} T^M h^{r_0} & \bmod N^s \text{ if } j = 0 \\ g^{-r_{j-1}} h^{r_j} & \bmod N^s \text{ if } j \in \{1, \ldots, d\} \\ g^{-r_d} & \bmod N^s \text{ if } j = d+1. \end{cases}$$

- $\mathsf{Dec}_{prm}(sk, C)$: Parse C as $(c_{d+1}, \ldots, c_0)$ and output

$$M \leftarrow L(c_0 {c_1}^x \cdots {c_{d+1}}^{x^{d+1}} \bmod N^s).$$

Above, L is the function given in Theorem 1.

Security: We will prove the following theorem in Section 6.

Theorem 4 (KDM Security of Our Scheme w.r.t $\mathcal{MAC}_{n,d,\ell}[N^{s-1}]$). *For any polynomial d, n, and ℓ of the security parameter κ, the proposed scheme is KDM secure with respect to $\mathcal{MAC}_{n,d,\ell}[N^{s-1}]$ under the DCR assumption.*

Specifically, for any polynomials n, d, and ℓ of κ, and any polynomial time adversary A for breaking KDM security of our scheme, there exists an adversary B for breaking the DCR problem in $\mathbb{Z}_{N^s}$ satisfying

$$\begin{cases} \mathsf{Adv.KDM}_{\mathsf{A}}[\mathcal{MAC}_{n,d,\ell}[N^{s-1}], n] \leq 6\mathsf{Adv.DCR}_{\mathsf{B}} + O\left(\frac{qd}{\sqrt{N}}\right) + O\left(\frac{n}{2^\xi}\right), \\ t_{\mathsf{B}} \leq t_{\mathsf{A}} + O(qdE) + O(q\ell d^2 \kappa^2) \end{cases}$$

Above, q is the number of queries of A, $t_{(\cdot)}$ is the number of steps of machines, and E is a full exponentiation cost in $\mathcal{QR}[N^s]$.

We can show a stronger variant of Theorem 4 where an adversary can select parameter d and ℓ *on the fly* when it makes encryption queries. (Specifically, d becomes encryption bounded and ℓ unbounded as indicated in Section 1.4.) The proof of this stronger security is similar to that of Theorem 4. We therefore omit it.

4 KDM Secure Scheme w.r.t. Fraction of Bounded Degree MACs

In this section, we give a general converter from a KDM[$\mathcal{F}$] secure scheme to a KDM[$\mathcal{Q}(\mathcal{F})$] secure scheme, where $\mathcal{F}$ is a set of polynomials over $\mathbb{Z}_K$ and $\mathcal{Q}(\mathcal{F})$

denote the set of all rational functions $f'(\overrightarrow{X})/f''(\overrightarrow{X})$ for $f, g \in \mathcal{F}$. By applying this converter to our first scheme, we can get a KDM[$\mathcal{Q}(\mathcal{MAC}_{n,d,\ell}[K])$] secure scheme.

A subtle but difficult problem in designing KDM[$\mathcal{Q}(\mathcal{F})$] secure scheme is that the denominator of $f'(\overrightarrow{sk})/f''(\overrightarrow{sk})$ can be 0 (or more generally, can be non-invertible): This becomes problem when proving security of the scheme because a simulator in the security proof (which does not know sk) cannot know whether $f''(\overrightarrow{sk})$ is invertible or not. We therefore have to design our scheme and prove the security of it such that a simulator can simulate the view of adversary even without knowing whether $f''(\overrightarrow{sk})$ is invertible.

We assume the hardness of factoring of the modulus K. Then no one can find value $a \in \mathbb{Z}_K$ which is non zero but is non-invertible. (If one can find such a, he can factorize K by computing $\mathsf{gcd}(a, K)$.) Therefore, we can assume that the value $f''(\overrightarrow{x})$ is either invertible or 0.

We then define the function value $f'(\overrightarrow{x})/f''(\overrightarrow{x})$ with $f''(\overrightarrow{x}) = 0$ as follows, where $\mathtt{1/0}$ and $\mathtt{0/0}$ are special symbols.

$$f(\overrightarrow{x}) = \begin{cases} \mathtt{1/0} & \text{if } f''(\overrightarrow{x}) = 0 \text{ but } f'(\overrightarrow{x}) \neq 0 \\ \mathtt{0/0} & \text{if } f''(\overrightarrow{x}) = f'(\overrightarrow{x}) = 0. \end{cases}$$

Note that we are not required to consider the other case (that is, $f''(x)$ is not 0 but is not invertible) due to the above discussion.

Let $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme whose secret key and message spaces are $\mathbb{Z}_K$ for some integer K. The scheme $\overline{\mathcal{PKE}} = (\overline{\mathsf{Setup}}, \overline{\mathsf{Kg}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ converted from $\mathcal{PKE}$ is as follows.

- The message space of $\overline{\mathcal{PKE}}$ is $\mathbb{Z}_K \cup \{\mathtt{1/0}, \mathtt{0/0}\}$. Here, "$\mathtt{1/0}$" and "$\mathtt{0/0}$" are special symbols.
- $\overline{\mathsf{Setup}}(1^\kappa)$ and $\overline{\mathsf{Kg}}(prm)$: The same as $\mathsf{Setup}(1^\kappa)$ and $\mathsf{Kg}(prm)$.
- $\overline{\mathsf{Enc}}_{prm}(pk, M)$: Select $R \overset{\$}{\leftarrow} \mathbb{Z}_K^*$ randomly and set

$$(M', M'') \leftarrow \begin{cases} (MR, R) \bmod K & \text{if } M \neq \mathtt{1/0}, \mathtt{0/0}, \\ (\ R,\ \ 0) \bmod K & \text{if } M = \mathtt{1/0}. \\ (\ 0,\ \ 0) \bmod K & \text{if } M = \mathtt{0/0}. \end{cases}$$

  Compute and output

$$\bar{C} \leftarrow (C', C'') \leftarrow (\mathsf{Enc}_{prm}(pk, M'), \mathsf{Enc}_{prm}(pk, M'')).$$

- $\overline{\mathsf{Dec}}_{prm}(sk, \bar{C})$: Parse $\bar{C}$ as (C', C''), compute

$$M' \leftarrow \mathsf{Dec}_{prm}(sk, C'); \qquad M'' \leftarrow \mathsf{Dec}_{prm}(sk, C'').$$

  Output $\mathtt{1/0}$ if $M' \neq 0$ and $M'' = 0$ holds. Output $\mathtt{0/0}$ if $M' = M'' = 0$ holds. Output $M \leftarrow M'/M'' \bmod K$ otherwise.

Theorem 5. *Suppose that factoring of K is hard. Suppose the following property also: for any $f(\overrightarrow{X}) \in \mathcal{F}$ and $R \in \mathbb{Z}_K$, the function $R \cdot f(\overrightarrow{X})$ is an element of $\mathcal{F}$. Then, KDM$[\mathcal{F}]$ security of $\mathcal{PKE}$ implies KDM$[\mathcal{Q}(\mathcal{F})]$ security of $\overline{\mathcal{PKE}}$.*

Proof. (sketch) An adversary B for KDM$[\mathcal{F}]$ security of $\mathcal{PKE}$ is constructed from an adversary A for KDM$[\mathcal{Q}(\mathcal{F})]$ security of $\overline{\mathcal{PKE}}$ as follows. B takes a public parameter *prm* and a tuple $(pk_j)_{j\in[n]}$ of public keys as an input and passes it to A. If A makes a query $i \in [n]$ and a pair (D', D'') of descriptions of MACs, B selects $S \overset{\$}{\leftarrow} \mathbb{Z}_K^*$ randomly, sets E' and E'' to the descriptions of functions $S \cdot f_{D'}(\overrightarrow{X})$ and $S \cdot f_{D''}(\overrightarrow{X})$ respectively, makes queries (i, E') and (i, E''), gets answers C' and C'' from the challenger, and sends (C', C'') back to A as an answer to the query. If A outputs a bit b', B outputs b'.

From the hardness of the factoring of K, the values $f_{D''}(\overrightarrow{sk})$ is either invertible or equal to 0. Hence, we can consider $1/f_{D''}(\overrightarrow{sk}) \bmod K$ if $f_{D''}(\overrightarrow{sk}) \neq 0$. Therefore,

$$(S\cdot f_{D'}(\overrightarrow{sk}), S\cdot f_{D''}(\overrightarrow{sk})) = \begin{cases} (\frac{f_{D'}(\overrightarrow{sk})}{f_{D''}(\overrightarrow{sk})} \cdot R_0, R_0) & \text{if } f_{D''}(\overrightarrow{sk}) \neq 0 \\ (R_1, 0) & \text{if } f_{D''}(\overrightarrow{sk}) = 0 \text{ but } f_{D'}(\overrightarrow{sk}) \neq 0 \\ (0, 0) & \text{if } f_{D'}(\overrightarrow{sk}) = f_{D''}(\overrightarrow{sk}) = 0, \end{cases}$$

where $R_0 = S \cdot f_{D''}(\overrightarrow{sk})$ and $R_1 = S \cdot f_{D'}(\overrightarrow{sk})$.

The message which B should encrypt in the above three cases is $f_{D'}(\overrightarrow{sk})/f_{D''}(\overrightarrow{sk})$, $\mathsf{1/0}$, and $\mathsf{0/0}$ respectively. This means that the view of A simulated by B is identical to the actual one.

The above proof does not work well if the factoring of K is easy, because A may make query (D', D'') such that $f_{D''}(sk)$ is not 0 but non-invertible. This means that K cannot be N^{s-1} for $s \geq 3$.

5 Triple Mode Proof Framework

5.1 Overview

A triple mode proof framework is introduced to overcome the dilemma described in Section 1.5. It has three modes called *standard mode*, *fake mode*, and *hiding mode*. The standard mode is the same as the original game of KDM security. Other two modes are as follows. (See Fig.2 also.)

Fake Mode: This mode allows us to compute "fake ciphertexts" using queries (i, D) of an adversary but without using the secret keys. The fake ciphertexts should be indistinguishable from the ciphertext of the standard mode, under the condition that a simulator *knows* the secret keys.

This indistinguishability of course cannot be proved based on the hardness related to unavailability of the secret keys. Instead, we are required to prove it

Dependency of Ciphertexts	sk	D	
Standard Mode	Yes.	Yes.	} Sim. knows sk.
Fake Mode	No.	Yes.	} Sim. does not know sk.
Hide Mode	No.	No.	

Fig. 2. Triple Mode Proof Framework

based on the secrecy of the *randomness of encryptions*. Showing this indistinguishability is the critical part of the proof of KDM security.

Hiding Mode: This mode enables us to compute "hiding ciphertexts" using neither the queries (i, D) of an adversary nor the secret keys. The hiding ciphertexts should be indistinguishable from the fake ones, under the assumption that a simulator *does not know* the secret keys. Note that the simulator is not required to know the secret keys in fact, because both the fake ciphertexts and the hiding ones can be computed without using the secret keys. This indistinguishability can be shown using the standard cryptographic arguments based on the secrecy of the secret key. Since the hiding ciphertext does not depend on the query D of an adversary, KDM security, in turn, clearly holds.

5.2 Formal Description

A *triple mode proof framework* for an encryption scheme $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ is a pair of *fake mode* $(\mathsf{KgFake}, \mathsf{EncFake})$ and *hiding mode* $(\mathsf{KgHide}, \mathsf{EncHide})$. The inputs and outputs of the algorithms are as follows.

- KgFake takes a public parameter prm and a natural number n as inputs and outputs n key pairs (pk_1, sk_1), ..., (pk_n, sk_n) and aux.
- KgHide takes the same inputs as KgFake and outputs keys $pk_1, \ldots, pk_n$ and aux.
- $\mathsf{EncFake}$ takes as inputs a public parameter prm, a tuple of public keys $(pk_j)_{j \in [n]}$, aux, a natural number i, and a description D of some function in $\mathcal{F}$ and outputs C.
- $\mathsf{EncHide}$ takes the same inputs as KgFake except D and outputs C.

A proof of KDM security of $\mathcal{PKE}$ w.r.t. a functions set $\mathcal{F}$ proceeds by showing that the success probability $\Pr[\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{F}, n] = 1]$ of A in $\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{F}, n]$ is the same as those in the following two games $\mathsf{GameFake}_{\mathsf{A}}[\mathcal{F}, n]$ and $\mathsf{GameHide}_{\mathsf{A}}[\mathcal{F}, n]$ but for a negligible differences.

- $\mathsf{GameFake}_{\mathsf{A}}[\mathcal{F}, n]$:

$$prm \leftarrow \mathsf{Setup}(1^\kappa), ((pk_j, sk_j)_{j \in [n]}, aux) \leftarrow \mathsf{KgFake}(prm, n),$$
$$b' \leftarrow \mathsf{A}^{\mathcal{O}'_{\mathsf{Kg}}, \mathcal{O}_{\mathsf{EncFake}_{prm}}}(prm, (pk_j)_{j \in [n]}), \text{ Output } b'$$

- $\mathsf{GameHide}_{\mathsf{A}}[\mathcal{F}, n]$:

$$prm \leftarrow \mathsf{Setup}(1^{\kappa}), ((pk_j)_{j\in[n]}, aux) \leftarrow \mathsf{KgHide}(prm, n),$$
$$b' \leftarrow \mathsf{A}^{\mathcal{O}'_{\mathsf{Kg}}, \mathcal{O}_{\mathsf{EncHide}_{prm}}}(prm, (pk_j)_{j\in[n]}), \text{ Output } b'$$

Above, A is allowed to make polynomial number of queries adaptively. It can send as queries bit string new to $\mathcal{O}'_{\mathsf{Kg}}$ and a tuple (i, D) to $\mathcal{O}_{\mathsf{EncFake}_{prm}}$ and to $\mathcal{O}_{\mathsf{EncHide}_{prm}}$. Here $i \in [n]$ is an integer and D is a description of some function in $\mathcal{F}$. The answers from the oracles are as follows:

- $\mathcal{O}'_{\mathsf{Kg}}(\mathsf{new})$ returns pk_i generated by KgFake (in $\mathsf{GameFake}$) or KgHide (in $\mathsf{GameHide}$).
- $\mathcal{O}_{\mathsf{EncFake}_{prm}}(i, D)$ returns $\mathsf{EncFake}_{prm}((pk_j)_{j\in[n]}, aux, i, D)$.
- $\mathcal{O}_{\mathsf{EncHide}_{prm}}(i, D)$ returns $\mathsf{EncHide}_{prm}((pk_j)_{j\in[n]}, aux, i)$.

In the final game, $\mathsf{GameHide}_{\mathsf{A}}[\mathcal{F}, n]$, ciphertexts do not depend on f queried by A any more. Hence, the following theorem holds.

Theorem 6. *Suppose that for any polynomial time adversary* A*, there exists a negligible function* $\varepsilon(\kappa)$ *such that the differences among* $\Pr[\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{F}] = 1]$*,* $\Pr[\mathsf{GameFake}_{\mathsf{A}}[\mathcal{F}, n] = 1]$*, and* $\Pr[\mathsf{GameHide}_{\mathsf{A}}[\mathcal{F}, n] = 1]$ *is less than* $\varepsilon(\kappa)$ *for any* $n = \mathsf{poly}(\kappa)$*, then the scheme is KDM*$[\mathcal{F}]$ *secure.*

As noted, proofs of known KDM secure scheme in the standard model [BHHO08,ACPS09,BG10] can be re-interpreted as above.

6 Security Proof of the First Scheme

6.1 Interactive Vector Lemma [BG10]

We review a lemma of [BG10] which we will use to prove KDM security of our scheme. Let $\mathsf{Gen}(1^{\kappa})$ be a generator which outputs the product N of two safe primes with the same bit lengths. Let A be a polynomial time adversary, b be a bit, and $s \geq 2$ be an integer. Define game IV_1 and IV_2 as follows.

- $\mathsf{IV}_1 : N \leftarrow \mathsf{Gen}(1^{\kappa}), g \overset{\$}{\leftarrow} \mathcal{SCR}[N^s], \quad b' \leftarrow \mathsf{A}^{\mathcal{O}_b}(s, N, g), \quad$ Output b'.
- $\mathsf{IV}_2 : N \leftarrow \mathsf{Gen}(1^{\kappa}),\ g, h \overset{\$}{\leftarrow} \mathcal{SCR}[N^s],\ b' \leftarrow \mathsf{A}^{\bar{\mathcal{O}}_b}(s, N, g, h), \quad$ Output b'.

In the above two games, A is allowed to make polynomial number queries. In IV_1, A can send an element δ of $\mathbb{Z}_{N^{s-1}}$ as a query. $\mathcal{O}_b(\delta)$ then selects $r \overset{\$}{\leftarrow} [\lfloor N/4 \rfloor]$ randomly and returns

$$u^* \leftarrow \begin{cases} T^{\delta} g^r \bmod N^s & \text{if } b = 1, \\ g^r \bmod N^s & \text{if } b = 0. \end{cases}$$

On the other hand, A in IV_2 can send an element $(\delta, \bar{\delta})$ of ${\mathbb{Z}_{N^{s-1}}}^2$ as a query. $\bar{\mathcal{O}}_b(\delta, \bar{\delta})$ then selects $r \overset{\$}{\leftarrow} [\lfloor N/4 \rfloor]$ randomly and returns

$$(u^*, \bar{u}^*) \leftarrow \begin{cases} (T^{\delta} g^r, T^{\bar{\delta}} h^r) \bmod N^s & \text{if } b = 1, \\ (g^r, h^r) \bmod N^s & \text{if } b = 0. \end{cases}$$

For $k = 1, 2$, the advantage of A in IV_k is defined to be

$$\mathsf{Adv.IV}_k[\mathsf{A}] = |\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]|.$$

Lemma 1 ((DCR-based) Interactive Vector Lemma for $k = 1, 2$, Full paper of [BG10]). *For $k = 1, 2$, no polynomial time adversary can have non-negligible advantage in IV_k under the DCR assumption.*

Our definition of game IV_k is slightly different from those of [BG10]: The original game takes the randomness r of g^r not from $[\lfloor N/4 \rfloor]$ but $[T^2]$ for some fixed value $T \geq N^s$. This difference is not essential, because the randomness r of the original game is taken from $[T^2]$ in order to be ensured that the distribution of g^r is statistically close to the uniform distribution on $\mathcal{SCR}[N^s]$. It can be shown that the same thing holds even if r is selected from $[\lfloor N/4 \rfloor]$.

6.2 The Proof when The Number n of Keys is 1.

Before proving the security of our scheme, the game $\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{MAC}]$ for our scheme with $n = 1$ is reviewed. (Here we simply write $\mathcal{MAC}$ for $\mathcal{MAC}_{n,d,\ell}[N^{s-1}]$.) An adversary A of this game takes a public parameter $prm = (s, N, g)$ and one public key $pk = h = g^x \bmod N^s$ as inputs. Whenever A sends as a query the "description" of a function in $\mathcal{F}$, namely an MAC D, the challenger sends back

$$C = ({u_d}^{-1}, {u_{d-1}}^{-1} v_d, \ldots, {u_0}^{-1} v_1, T^{f_D(x)} v_0),$$

where f_D is the function corresponding to D and $(u_k, v_k) = (g^{r_k}, h^{r_k})$ for $r_k \leftarrow \lfloor N/4 \rfloor$. Since the number of keys is 1, the polynomial f_D can be written as $f_D(Y) = \sum_j a_j Y^j \bmod N^s$ for some $(a_j)_{j \in [0..d]}$. A finally outputs a bit b'.

The security is proved based on the framework of triple mode proof framework of Section 5. The algorithms KgFake and $\mathsf{EncFake}$ are defined as follows.

- $\mathsf{KgFake}(prm)$: Same algorithm as Kg, except that it sets aux to the null string.
- $\mathsf{EncFake}(prm, pk, aux, D)$: Parse prm and pk as (s, N, g) and h. Take $r_k \overset{\$}{\leftarrow} \lfloor N/4 \rfloor$ and compute $(u_k, v_k) \leftarrow (g^{r_k}, h^{r_k})$ for $k \in [d]$. Let $f_D(Y) = \sum_j a_j Y^j \bmod N^s$. Compute and output a "fake ciphertext"

$$C_{\mathsf{Fake}} = ({u_d}^{-1}, T^{a_d} {u_{d-1}}^{-1} v_d, \ldots, T^{a_1} {u_0}^{-1} v_1, T^{a_0} v_0).$$

We show that $|\Pr[\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{MAC}, 1] = 1] - \Pr[\mathsf{GameFake}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]|$ is negligible. To this end, an adversary B for the game IV_k with $k = 1$ is constructed as follows. B takes an input (s, N, g), selects $x \overset{\$}{\leftarrow} [2^{\xi} \cdot \lfloor N/4 \rfloor]$ randomly, and feeds $prm \leftarrow (s, N, g)$ and $pk \leftarrow h \leftarrow g^x$ to A. If A makes a query D, B computes $(a_j)_{j \in [0..d]}$ satisfying $f_D(Y) = \sum_j a_j Y^j \bmod N^s$. Note that B can compute it in polynomial time from D. B sets

$$\delta_j = - \sum_{k=j+1}^{d} a_k x^{k-(j+1)},$$

makes queries $\delta_0, \ldots, \delta_{d-1}$ and gets corresponding answers $u_0^*, \ldots, u_{d-1}^*$ (where u_j^* is $T^{\delta_j} g^{r_j}$ or g^{r_j}). B then selects $r_d \leftarrow [\lfloor N/4 \rfloor]$, computes $u_d^* \leftarrow g^{r_d}$, computes $v_j^* \leftarrow (u_j^*)^x$ for $j = 0, \ldots, d$, and sends back to A

$$C^* = ((u_d^*)^{-1}, (u_{d-1}^*)^{-1} v_d^*, \ldots, (u_0^*)^{-1} v_1^*, T^{f_D(x)} v_0^*).$$

If A outputs a bit b', B outputs it and terminates. From the definition of B, the difference between two probabilities $\Pr[\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]$ and $\Pr[\mathsf{GameFake}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]$ is negligible.

The algorithms in $\mathsf{GameHide}$, that is KgHide and $\mathsf{EncHide}$, are defined as follows.

- $\mathsf{KgHide}(prm)$: Take $pk \leftarrow h \stackrel{\$}{\leftarrow} \mathcal{QR}[N^s]$ and outputs it. (It sets aux to the null string.)
- $\mathsf{EncHide}(prm, pk, aux)$: Parse prm and pk as (s, N, g) and $pk = h$. Take $r_k \stackrel{\$}{\leftarrow} \lfloor N/4 \rfloor$ and compute $(u_k, v_k) \leftarrow (g^{r_k}, h^{r_k})$ for $k \in [d]$. Compute and output a "hiding ciphertext"

$$C_{\mathsf{Hide}} = ({u_d}^{-1}, {u_{d-1}}^{-1} v_d, \ldots, {u_0}^{-1} v_1, v_0).$$

Namely, $\mathsf{EncHide}$ outputs $\mathsf{Enc}_{prm}(pk, 0)$.

We show that $|\Pr[\mathsf{GameFake}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1] - \Pr[\mathsf{GameHide}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]|$ is negligible. To this end, an adversary B for IV_k with $k = 2$ is constructed as follows. B takes an input (s, N, g, h), and feeds $prm \leftarrow (s, N, g)$ and $pk \leftarrow h$ to A. If A makes a query D, let $f_D(Y) = \sum_j a_j Y^j \bmod N^s$. B then sets

$$(\delta_j, \bar{\delta}_j) = (0, a_j)$$

makes queries $(\delta_0, \bar{\delta}_0), \ldots, (\delta_d, \bar{\delta}_d)$ and gets answers $(u_1^*, v_1^*), \ldots, (u_d^*, v_d^*)$ (where (u_j^*, v_j^*) is $(T^0 g^{r_j}, T^{a_j} h^{r_j})$ or (g^{r_j}, h^{r_j})) and sends back to A

$$C^* = ((u_d^*)^{-1}, (u_{d-1}^*)^{-1} v_d^*, \ldots, (u_0^*)^{-1} v_1^*, v_0^*).$$

If A outputs a bit b', B outputs it and terminates. From the definition of B, the difference between two probabilities $\Pr[\mathsf{GameFake}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]$ and $\Pr[\mathsf{GameHide}_{\mathsf{A}}[\mathcal{MAC}, 1] = 1]$ is negligible. From Theorem 6, our scheme is KDM secure.

6.3 The Idea Behind the Proof of the General Case

Due to the lack of space, we only present the proof idea. It proceeds in a similar way to the proof of Section 6.2, except that we make KgFake and $\mathsf{EncFake}$ "reduce" n secrets $(sk_j)_{j \in [n]}$ to only one secret μ.

Specifically, KgFake for this proof takes $prm = (s, N, g)$ and the number n of keys as inputs, selects $\mu \stackrel{\$}{\leftarrow} [\lfloor N/4 \rfloor]$ and $\alpha_1, \ldots, \alpha_n \stackrel{\$}{\leftarrow} [2^{\xi} \cdot \lfloor N/4 \rfloor]$ randomly, and outputs

$$sk_j \leftarrow x_j \leftarrow \mu + \alpha_j, \quad pk_j \leftarrow h_j \leftarrow g^{x_j} \text{ for } j \in [n], \quad aux \leftarrow (\alpha_j)_{j \in [n]}$$

In other words, $(sk_j)_{j\in[n]}$ is computed from only one "secret" μ. The proof is therefore reduced to the case where the number of secret is 1, in some sense.

The description of $\mathsf{EncFake}$ is also changed, in order to become consistent with the new KgFake. Specifically, $\mathsf{EncFake}$ takes $prm = (s, N, g)$, $(pk_j)_{j\in[n]} = (h_j)_{j\in[n]}$, and a query (i, D) of an adversary and computes $(a_j)_{j\in[0..d]} \leftarrow \mathsf{Coeff}_{prm}(aux, i, D)$. Here $(a_j)_{j\in[0..d]} = \mathsf{Coeff}_{prm}(aux, i, D)$ is the tuple of $\mathbb{Z}_{N^{s-1}}$ satisfying the following equations about polynomials of a variable Y. Below, d is the total degree of f_D.

$$f_D(Y + \alpha_1, \dots, Y + \alpha_n) = \sum_{j=0}^{d} a_j (Y + \alpha_i)^j \bmod N^{s-1}.$$

$\mathsf{EncFake}$ then outputs "fake encryption"

$$C_{\mathsf{Fake}} = ({u_d}^{-1}, T^{a_d}{u_{d-1}}^{-1}v_d, \dots, T^{a_1}{u_0}^{-1}v_1, T^{a_0}v_0),$$

where $(u_k, v_k) = (g^{r_k}, {h_i}^{r_k})$ for $r_k \overset{\$}{\leftarrow} \lfloor N/4 \rfloor$. (We stress that v_k is computed using the i-th public key h_i where i is a part of the query (i, D).)

Above, $\mathsf{EncFake}$ is polynomial time algorithm due to the following lemma:

Lemma 2. *Given $aux = (\alpha_j)_{j\in[n]}$, $i \in [n]$, and an MAC D,* $\mathsf{Coeff}_{prm}(aux, i, D)$ *can be computed in polynomial time.*

KgHide and $\mathsf{EncHide}$ can be constructed similarly. Proofs of indistinguishabilities of $\mathsf{GameKDM}^1_{\mathsf{A}}[\mathcal{MAC}, n]$, $\mathsf{GameFake}_{\mathsf{A}}[\mathcal{MAC}, n]$, and $\mathsf{GameHide}_{\mathsf{A}}[\mathcal{MAC}, n]$ are similar to those of Section 6.2 as well.

Acknowledgments. We thank Zvika Brakerski and Yevgeniy Vahlis for several illuminating discussions. We also thank Zvika, Shafi Goldwasser, Yael Tauman Kalai, and anonymous Eurocrypt reviewers for helpful comments regarding the presentation of the paper.

References

[AR00] Martín Abadi, Phillip Rogaway: Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). IFIP TCS 2000: 3-22, J. Cryptology 15(2): 103-127 (2002), J. Cryptology 20(3): 395 (2007)

[ABBC10] Tolga Acar and Mira Belenkiy and Mihir Bellare and David Cash RYPTO Cryptographic Agility and its Relation to Circular Encryption. EUROCRYPT 2010: 403-422

[ABHS05] Pedro Adão, Gergei Bana, Jonathan Herzog, Andre Scedrov: Soundness of Formal Encryption in the Presence of Key-Cycles. ESORICS 2005: 374-396

[A11] Benny Applebaum Key-Dependent Message Security: Generic Amplification and Completeness Theorems. EUROCRYPT 2011.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, Amit Sahai: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. C 2009: 595-618

[BDU08] Michael Backes, Markus Dürmuth, Dominique Unruh: OAEP Is Secure under Key-Dependent Messages. ASIACRYPT 2008: 506-523

[BPS08] Michael Backes, Birgit Pfitzmann, Andre Scedrov: Key-dependent message security under active attacks - BRSIM/UC-soundness of Dolev-Yao-style encryption with key cycles. CSF 2007: 112-124, Journal of Computer Security 16(5): 497-530 (2008).

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, Yuval Ishai: Bounded Key-Dependent Message Security. EUROCRYPT 2010: 423-444

[BRS02] John Black, Phillip Rogaway, Thomas Shrimpton: Encryption-Scheme Security in the Presence of Key-Dependent Messages. Selected Areas in Cryptography 2002: 62-75

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and Leakage Resilient Public-Key Encryption Under Subgroup Indistinguishability (or: Quadratic Residuosity Strikes Back). CRYPTO 2010: 1-20. Full paper is available at eprint 2010/226

[BGK09] Zvika Brakerski, Shafi Goldwasser, and Yael Kalai. Circular-Secure Encryption Beyond Affine Functions. e-print. 2009/511.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, Rafail O: Circular-Secure Encryption from Decision Diffie-Hellman. CRYPTO 2008: 108-125

[BV98] Dan Boneh, Ramarathnam Venkatesan: Breaking RSA May Not Be Equivalent to Factoring. EUROCRYPT 1998: 59-71

[CCS09] Jan Camenisch, Nishanth Chandran, Victor Shoup: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. EUROCRYPT 2009: 351-368

[CL01] Jan Camenisch, Anna Lysyanskaya: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. EUROCRYPT 2001: 93-118

[CKVW10] Ran Canetti, Yael Tauman Kalai, Mayank Varia, Daniel Wichs: On Symmetric Encryption and Point Obfuscation. TCC 2010: 52-71

[DJ01] Ivan Damgård, Mads Jurik: A Generalization, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. Public Key Cryptography 2001: 119-136

[G09] Craig Gentry: Fully homomorphic encryption using ideal lattices. STOC 2009: 169-178

[GH10] Matthew Green and Susan Hohenberger. CPA and CCA-Secure Encryption Systems that are not 2-Circular Secure. e-print. 2010/144

[HH09] Iftach Haitner, Thomas Holenstein: On the (Im)Possibility of Key Dependent Encryption. TCC 2009: 202-219

[HK07] Shai Halevi, Hugo Krawczyk: Security under key-dependent inputs. ACM CCS 2007: 466-475

[HU08] Dennis Hofheinz, Dominique Unruh: Towards Key-Dependent Message Security in the Standard Model. EUROCRYPT 2008: 108-126

[KTY09] Aggelos Kiayias, Yiannis Tsiounis, Moti Yung: Group Encryption. ASIACRYPT 2007: 181-199

[MTY11] Tal Malkin, Isamu Teranishi, Moti Yung. Key Dependent Message Security: Recent Results and Applications. ACM CODASPY 2011.

[P99] Pascal Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT 1999, 223-238.