Circular and Leakage Resilient Public-Key Encryption Under Subgroup Indistinguishability

(or: Quadratic Residuosity Strikes Back)

Zvika Brakerski[1] and Shafi Goldwasser[2]

[1] Weizmann Institute of Science
zvika.brakerski@weizmann.ac.il
[2] Weizmann Institute of Science and Massachusetts Institute of technology
shafi@theory.csail.mit.edu

Abstract. The main results of this work are new public-key encryption schemes that, under the quadratic residuosity (QR) assumption (or Paillier's decisional composite residuosity (DCR) assumption), achieve key-dependent message security as well as high resilience to secret key leakage and high resilience to the presence of auxiliary input information. In particular, under what we call the *subgroup indistinguishability assumption*, of which the QR and DCR are special cases, we can construct a scheme that has:

- **Key-dependent message (circular) security.** Achieves security even when encrypting affine functions of its own secret key (in fact, w.r.t. affine "key-cycles" of predefined length). Our scheme also meets the requirements for extending key-dependent message security to broader classes of functions beyond affine functions using previous techniques of Brakerski et al. or Barak et al.
- **Leakage resiliency.** Remains secure even if any adversarial low-entropy (efficiently computable) function of the secret key is given to the adversary. A proper selection of parameters allows for a "leakage rate" of $(1 - o(1))$ of the length of the secret key.
- **Auxiliary-input security.** Remains secure even if any sufficiently *hard to invert* (efficiently computable) function of the secret key is given to the adversary.

Our scheme is the first to achieve key-dependent security and auxiliary-input security based on the DCR and QR assumptions. Previous schemes that achieved these properties relied either on the DDH or LWE assumptions. The proposed scheme is also the first to achieve leakage resiliency for leakage rate $(1-o(1))$ of the secret key length, under the QR assumption. We note that leakage resilient schemes under the DCR and the QR assumptions, for the restricted case of composite modulus product of safe primes, were implied by the work of Naor and Segev, using hash proof systems. However, under the QR assumption, known constructions of hash proof systems only yield a leakage rate of $o(1)$ of the secret key length.

1 Introduction

The "classical" definition of *semantic secure* public-key encryption by Goldwasser and Micali [16], requires that an efficient attacker with access to the public encryption-key must not be able to find two messages such that it can distinguish a random encryption of one from a random encryption of the other. Numerous candidate public-key encryption schemes that meet this definition have been presented over the years, both under specific hardness assumptions (like the hardness of factoring) and under general assumptions (such as the existence of injective one-way trapdoor functions).

This notion of security, however (as well as other commonly accepted ones), does not capture certain situations that may occur in the "real world":

- Functions of the secret decryption-key can be encrypted and sent (note that semantic security only guarantees security with respect to messages which an efficient attacker can find).
- Information about the secret key may leak.
- The same secret key may be used in more than one application, or more generally the attacker can somehow obtain the value of a hard-to-invert function of the secret key.

In recent years, extensive research effort has been invested in providing encryption schemes which are provably secure even in the above settings. Such schemes are said to achieve *key-dependent message (*KDM*) security*, *leakage-resilience*, and *auxiliary-input security* in correspondence to the above real world settings. To date, we know of: (1) Candidate schemes which are KDM secure under the decisional Diffie-Hellman (DDH) and under the learning with errors (LWE) assumptions; (2) Candidate schemes that are resilient to key leakage of rate $(1-o(1))$ (relative to the length of the secret key), under the LWE assumption and under the DDH assumption. In addition, candidate scheme achieving some leakage resilience under a general assumption: the existence of universal hash-proof systems, with a leakage rate depending on the hash proof system being used; (3) Candidate schemes that are auxiliary input secure under the DDH assumption and under the LWE assumption.

In this work, we present an encryption scheme that achieves all of the above security notions simultaneously and is based on a class of assumptions that we call *subgroup indistinguishability assumptions*. Specifically, this class includes the quadratic residuosity (QR) and the decisional composite residuosity (DCR) assumptions, both of which are related to the problem of factoring large numbers. In addition, our schemes have the following interesting property: the secret key consists of a randomly chosen binary vector independent of the group at hand. The instantiation of our scheme under QR enjoys the same useful properties for protocol design as the original [16] scheme, including re-randomization of ciphertexts and support of the XOR homomorphic operation over the $\{0,1\}$ message space, with the added benefit of leakage resilience.

To best describe our results, we first, in Section 1.1, describe in detail the background for the new work, including the relevant security notions and pre-

vious results. Second, in Section 1.2, we describe in detail the new results and encryption schemes. Then, in Section 1.3, we describe the new techniques. Section 1.4 discusses some additional related works and Section 1.5 contains the paper organization.

1.1 Background

Key-dependent messages. The shortcoming of the standard security definition in the case where the plaintext to be encrypted depends on the secret key was already noticed in [16]. It was later observed that this situation is not so unlikely and may sometimes even be desirable [9, 1, 21]. Black, Rogoway and Shrimpton [5] formally defined KDM-security: the attacker can obtain encryptions of (efficient) functions of its choosing, taken from some specified class of functions $\mathcal{F}$, applied to the secret key. The requirement is that the attacker cannot tell if all of its queries are answered by encryptions of some constant symbol 0, instead of the requested values. This definition is extended to the case of many (say n) users that can encrypt each others' secret keys: the attacker's queries now contain a function to be applied to *all* secret keys, and an identity of the user whose public key should be used to encrypt. This latter case is referred to as KDM$^{(n)}$-security while the single-user case is called KDM$^{(1)}$-security.

Boneh, Halevi, Hamburg and Ostrovsky [6] constructed a public key encryption scheme that is KDM$^{(n)}$ secure w.r.t. all affine functions,[3] under the decisional Diffie-Hellman (DDH) assumption, for any polynomial n. This first result was followed by the work of Applebaum, Cash, Peikert and Sahai [3] who proved that a variation of Regev's scheme [25] is also KDM secure w.r.t. all affine functions, under the learning with errors (LWE) assumption.

More recent works by Brakerski, Goldwasser and Kalai [8] and by Barak, Haitner, Hofheinz and Ishai [4] presented each general and different techniques to extend KDM-security to richer classes of functions. In [8], the notion of *entropy-κ* KDM-security is introduced. A scheme is entropy-κ KDM-secure if it remains KDM-secure even if the secret key is sampled from a high-entropy distribution, rather than a uniform one. They show that an entropy-κ KDM-secure scheme implies a scheme that is KDM-secure w.r.t. roughly any pre-defined set of functions of polynomial cardinality. In [4], the notion of *targeted public-key encryption* is introduced. A targeted encryption scheme can be thought of as a combination of oblivious transfer and encryption: it is possible to encrypt in such a way that the ciphertext is decryptable only if a certain bit of the secret key takes a predefined value. They show that a targeted encryption scheme implies a KDM-secure scheme w.r.t. all functions computable by circuits of some predefined (polynomial) size. These two results achieve incomparable performance. While in the former, the public key and ciphertext lengths depend on the size of the function class (but not on its complexity) and are independent of the number of users n, in the latter the public key size does not depend on the function class, but

[3] More precisely "affine in the exponent": the secret key is a vector of group elements $g_1, \ldots, g_\ell$ and the scheme is secure w.r.t. functions of the form $h \cdot \prod g_i^{a_i}$.

the ciphertext length is linear in the product of n times the complexity of the functions.

Leakage resiliency. The work on cold boot attacks by Halderman et al. [17], gave rise to the notion of public-key encryption resilient to (bounded) memory leakage attacks, presented by Akavia, Goldwasser and Vaikuntanathan [2] and further explored by Naor and Segev [22]. In their definition, security holds even if the attacker gets some information of its choosing (depending on the value of the public key) on the scheme's secret key, so long as the total amount of information leaked does not exceed an a-priori information theoretic bound. More formally, the attacker can request and receive $f(sk)$ for a length-restricted function f.[4] [2, 22] presented public-key encryption schemes that are resilient to leakage of even a $1 - o(1)$ fraction of the secret key (we call this the "leakage rate"). In particular, [2] showed how this can be achieved under the LWE assumption, while [22] showed that this can be achieved under the DDH (or d-linear) assumption. It is further shown in [22] that some leakage resilience can be achieved using any universal hash proof system (defined in [10]), where the leakage rate depends on the parameters of the hash proof system. This implies secure schemes under the the QR and DCR assumptions as well. However, using the known hash proof systems, the leakage rate achievable under the QR assumption was only $o(1)$ — much less than the desired $1 - o(1)$. Based on the DCR assumption, a leakage rate of $(1 - o(1))$ was achievable [22, 10, 11].

Auxiliary input. Dodis, Kalai and Lovett [13] and Dodis, Goldwasser, Kalai, Peikert and Vaikuntanathan [12] considered the case where the leakage is not restricted information theoretically, but rather *computationally*. In the public key setting, the attacker is allowed to access any information on the secret key, with the following computational restriction: as long as recovering the secret key sk from said information $f(pk, sk)$, for f of the attackers choosing, is computationally hard to a sufficient extent (see discussion of several formalizations in [12]). This notion of security was termed *security in the presence of auxiliary input* (or *auxiliary-input security*, for short). Public-key auxiliary-input secure encryption schemes under the DDH and LWE assumptions were recently presented in [12].

1.2 New Results

Let us define a generalized class of assumptions called *subgroup indistinguishability* (SG) assumptions. A subgroup indistinguishability problem is defined by a group $\mathbb{G}_U$ ("the universe group") which is a direct product of two groups $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ (interpreted as "the group of messages" and "the language group") whose orders, denoted by M, L respectively, are relatively prime and where $\mathbb{G}_M$ is a cyclic group. Essentially, the *subgroup indistinguishability assumption* is that a random element of the universe $\mathbb{G}_U$ is computationally indistinguishable from a random element in $\mathbb{G}_L$. In other words, the language $\mathbb{G}_L$

[4] To be more precise, the requirement is that the min-entropy of the secret sk drops by at most a bounded amount, given $f(sk)$.

is hard on average in the universe $\mathbb{G}_U$. The precise definition is a little more involved, see Section 3 for details.

Two special cases of the subgroup indistinguishability assumptions are the quadratic residuosity (QR) assumption on Blum integers and Paillier's decisional composite residuosity (DCR) assumption. This is easily seen for QR as follows. Let integer $N = p \cdot q$, where p, q are random primes of equal bit-length, $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N : \gcd(x, N) = 1\}$, $\mathbb{J}_N$ denote the group of Jacobi symbol $(+1)$ elements of $\mathbb{Z}_N^*$, and $\mathbb{QR}_N = \{x^2 : x \in \mathbb{Z}_N^*\}$ denote its subgroup of quadratic residues. The *quadratic residuosity* (QR) assumption is then, that the uniform distributions over $\mathbb{J}_N$ and $\mathbb{QR}_N$ are computationally indistinguishable. Taking N to be a *Blum integer* where $p, q = 3 \pmod 4$ (otherwise the orders of $\mathbb{G}_L, \mathbb{G}_M$ we define next will not be relatively prime) and setting $\mathbb{G}_U = \mathbb{J}_N$, $\mathbb{G}_L = \mathbb{QR}_N$ (which is of odd order), and $\mathbb{G}_M = \{\pm 1\}$ (which is cyclic and has order 2), the QR assumption falls immediately into the criteria of subgroup indistinguishability assumptions.

We are now ready to describe the new encryption scheme for a given subgroup problem $(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L)$ where h is a generator for $\mathbb{G}_M$. In general, we view the *plaintext message space* as the elements $h^m \in \mathbb{G}_M$ (sometimes the exponent m itself can be viewed as the message). For the case of QR, the plaintext message space is $\mathbb{G}_M = \{\pm 1\}$.

A word on the choice of parameters is in order. All parameters are measured as a function of the security parameter k. As customary, in the QR and DCR cases, think of the security parameter as the size of the modulus N (i.e. $k = \lceil \log N \rceil$). We let ℓ denote a parameter whose value is polynomially related to k,[5] selected in accordance to the desired properties of the scheme (KDM security, amount of leakage resilience etc.).

The Encryption Scheme for Subgroup Problem $(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L)$ with Parameter ℓ:

- *Key generation.* Set the secret key to a random binary vector $\mathbf{s} = (s_1, \ldots, s_\ell)$ of length ℓ. Set the public key to be the tuple $(g_1, \ldots, g_\ell, g_0)$ where $g_1, \ldots, g_\ell$ are uniformly chosen elements of $\mathbb{G}_L$ and $g_0 = \prod g_i^{-s_i}$. (For the QR assumption, the public key thus consists of ℓ random squares, followed by a product of a random subset of them, selected by the secret key $\mathbf{s}$).
- *Encryption.* On input message h^m,[6] sample a uniform integer r from a large enough domain and output the ciphertext $(g_1^r, \ldots, g_\ell^r, h^m \cdot g_0^r)$. (For the QR assumption case, encryption is of single bits $\{\pm 1\}$, and the ciphertext is the tuple of squares in the public key, raised to a random power, where the last one is multiplied by the plaintext message.)
- *Decryption.* On ciphertext $(c_1, \ldots, c_\ell, c_0)$, compute $h^m = c_0 \cdot \prod c_i^{s_i}$. (For the case of QR, $m = c_0 \cdot \prod c_i^{s_i}$.) In general, recoverability of the exponent m depends on whether taking discrete logs in base h of h^m is easy.

[5] More precisely, ℓ is a polynomial function $\ell(k)$.

[6] Recall that h is a generator of $\mathbb{G}_M$, which is a part of the description of $\mathbb{G}_U$.

We remark that the basic structure of our construction is strikingly similar to [6], where the public key also contains ℓ independent "random" elements and an additional element that is statistically close to uniform, but in fact is a combination of the previous ones. The difference and challenge is in how to prove security. This challenge is due to the fact that the subgroup indistinguishability assumptions seem inherently different from the DDH assumption. In the latter, for cyclic group $\mathbb{G}$ where DDH is assumed, the assumption implies that the distribution (g_1, g_2, g_1^r, g_2^r) is computationally indistinguishable from (g_1, g_2, g_1', g_2') giving complete re-randomization (a similar property follows for LWE). Such re-randomization does not follow nor is it necessarily true from subgroup indistinguishability. Rather, we will have to use the weaker guarantee that (g_1, g_2, g_1^r, g_2^r) is indistinguishable from $(g_1, g_2, h^{r'} \cdot g_1^r, h^{r''} \cdot g_2^r)$, giving only "masking" of the message bits.

Similarly to the scheme of [6], our scheme is lacking in efficiency. This is most noticeable in our QR-based scheme, where the encryption of one bit requires a ciphertext containing $\ell + 1$ group elements, each of size roughly the security parameter k. The situation is somewhat better when relying on DCR: there each such ciphertext encrypts $\Omega(k)$ bits. Improved efficiency can be achieved by using the same values $g_1, \ldots, g_\ell$ with many vectors $\mathbf{s}$, however this makes KDM security hold only with respect to a less natural function class (this is similar to the more efficient LWE based scheme of [3]) and significantly reduces leakage resiliency. Coming up with more efficient KDM secure or leakage resilient schemes remains an interesting open problem.

We prove the following properties for the new encryption scheme.

Property 1: KDM-Security First, we prove that the scheme is $\text{KDM}^{(1)}$-secure w.r.t. affine functions of the secret key. To show this for QR case, we show that for any affine function specified by $a_0, \ldots, a_\ell$, the encryption of $(-1)^{a_0 + \sum_i a_i s_i}$ is indistinguishable from the encryption of $(-1)^0$. For the general case, it is more natural to view $\text{KDM}^{(1)}$ with respect to the affine functions "in the exponent": for any $h_0, h_1, \ldots, h_\ell \in \mathbb{G}_M$ where $h_i = h^{a_i}$, for the generator h, we show that an encryption of $h_0 \cdot \prod h_i^{s_i} = h^{a_0 + \sum_i a_i s_i}$ is indistinguishable from an encryption of h^0.

Second, we prove that for any polynomial value of n, the above encryption scheme satisfies $\text{KDM}^{(n)}$ security, if ℓ is larger than, roughly, $n \log L$. We note thus that the public key size and ciphertext size grow with n to achieve provable $\text{KDM}^{(n)}$ security. Interestingly, in the works of [6, 3], ℓ did not need to grow with n. This seems difficult to achieve without the complete "re-randomization" property discussed above which does follow from the DDH and LWE assumptions, but not from ours.

Finally, we can also show that our scheme can be used to obtain KDM security for larger classes of functions than affine function: The scheme is *entropy-κ KDM-secure* (for proper values of ℓ), as required in [8] and therefore implies a scheme that is secure w.r.t. functions of the form $a_0 + \sum_i a_i f_i(sk)$ for (roughly) any set of polynomially-many efficiently computable functions $\{f_1, \ldots, f_\ell\}$. Our scheme

also implies a *targeted encryption scheme*, as required in [4], and therefore implies that for any polynomial bound p, there is a scheme that is secure w.r.t. all functions computable by size-p circuits.

Property 2: Improved Key-Leakage Resiliency We prove that the new scheme is resilient to any leakage of a $(1-o(1))$ fraction of the bits of the secret key. Stated differently, if one specifies in advance the amount of leakage λ (a polynomial in the security parameter) to be tolerated, we can choose ℓ to obtain a scheme that is secure against a leakage of λ bits. The growth of ℓ is additive in λ (i.e. $\ell = \ell_0 + \lambda$) and therefore we can select the value of ℓ to obtain schemes that are resilient to leakage of a $(1-(\ell_0/\ell)) = (1-o(1))$ fraction of the secret key.

We emphasize that while schemes with the above guarantees were known under LWE [2] or DDH [22], and even (implicitly) under DCR [22, 10], this was not the case under QR. Previous results with regards to QR-based leakage resiliency [22, 10] could only approach a leakage rate of $1/k = o(1)$ (recall that k is the security parameter, or the bit-length of the modulus N), compared to $(1-o(1))$ in our scheme.

In addition, previous constructions of QR and DCR based hash proof systems required that the modulus used $N = p \cdot q$ is such that p, q are *safe primes*. We do not impose this restriction. In the QR case we only require that $p, q = 3 \pmod 4$ (i.e. N is a *Blum integer*) and in the DCR case we only require that p, q have the same bit-length.

Property 3: Auxiliary Input Security We prove that our schemes remain secure when the attacker has access to additional information on the secret key sk, in the form of $f_{pk}(sk)$, where f_{pk} is a polynomial time function (which may depend on the public key) that is evaluated on the secret key sk. First, we consider the case where f is such that the transition $(f_{pk}(sk), pk) \rightarrow sk$ is computationally hard. Namely, that retrieving the secret key sk given the public key pk and the auxiliary information $f_{pk}(sk)$, is sufficiently hard. This notion was termed *weak auxiliary-input* security in [12]. In turn, [12] show how to leverage weak auxiliary-input security to achieve security when the requirement on f is weaker: now, only the transition $f_{pk}(sk) \rightarrow sk$ needs to be hard. The latter is called *auxiliary-input* security.

We conclude that for all $\delta > 0$, we can select the value of ℓ such that the scheme is auxiliary-input secure relative to any function that is hard to invert (in polynomial time) with probability $2^{-\ell^\delta}$. We note that the input to the function is the secret key – a length ℓ binary string, and therefore we measure hardness as a function of ℓ (and not of the security parameter k).

1.3 Our Techniques

The circular security, leakage resiliency and auxiliary-input security properties of our scheme are proved using a new technical tool introduced in this work:

the *interactive vector game*. This proof technique can also provide an alternative proof for the KDM$^{(1)}$-security, leakage resiliency and auxiliary-input security of (known) public-key encryption schemes based on DDH and LWE, thus providing an alternative, more generic proof for some of the results of [6, 3, 22, 12].[7]

This suggests an alternative explanation to the folklore belief that the three notions are related: that it is the proof technique that is related in fact. Namely, the proof techniques for each property can be generalized to interactive vector games which, in turn, imply the other properties.

We proceed to overview the proofs of security for the various properties of our scheme. Again, let us consider the groups $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ with h being a generator for $\mathbb{G}_M$, such that the subgroup indistinguishability assumption holds.

To best explain the ideas of the proof, let us consider, as a first step, a simple semantically secure encryption scheme (which is a generalization of the Goldwasser-Micali scheme [15]). An encryption of 0 is a random element $g \in \mathbb{G}_L$ and an encryption of 1 is $h \cdot g$ (in the QR case, the encryption of $(+1)$ is a random quadratic residue and the encryption of (-1) is a random quadratic non-residue). The two distributions are clearly indistinguishable (consider the indistinguishable experiment where g is uniform in $\mathbb{G}_U$). In order to decrypt, one needs some "trapdoor information" that would enable to distinguish between elements in $\mathbb{G}_L$ and $\mathbb{G}_U$ (such as the factorization of the modulus N in the QR (and DCR) case).

The first modification of this simple idea was to fix g and put it in the public key, and set the ciphertext for h^m to $h^m \cdot g^r$ for r large enough. Note that the sender does not know the order of $\mathbb{G}_U$: Indeed, in the QR case, knowing the order of the group $\mathbb{J}_N$, which is $\frac{\varphi(N)}{2}$, enables to factor N. For the QR case, this modification still amounts to encrypting $(+1)$ by a random square, and (-1) by a random non-square.

The second modification does away with the need of the secret key owner to distinguish between elements in $\mathbb{G}_L$ and $\mathbb{G}_U$ (e.g. with the need to know the factorization of N in the QR case), by replacing the "trapdoor information" with a secret key that is a uniform binary vector $\mathbf{s} = (s_1, \ldots, s_\ell)$. Holding the secret key will not enable us to solve subgroup indistinguishability, but will enable us to decrypt as in [6]. We take a set of random elements $g_1, \ldots, g_\ell \in \mathbb{G}_L$ and define $g_0 = \prod g_i^{-s_i}$. If ℓ is large enough, then the leftover hash lemma implies that g_0 is almost uniform. As the ciphertext is $(g_1^r, \ldots, g_\ell^r, h^m \cdot g_0^r)$, one can recover h^m using $\mathbf{s}$. Recovering m itself is also possible if the discrete logarithm problem in $\mathbb{G}_M$ is easy, as is the case in the QR scenario.

The crux of the idea in proving security is as following. First, we note that the distribution of g_0 is close to uniform in $\mathbb{G}_L$, even given $g_1, \ldots, g_\ell$ (by the leftover hash lemma). Recall that in a DDH-based proof, we could claim that $((g_1, \ldots, g_\ell, g_0), (g_1^r, \ldots, g_\ell^r, g_0^r))$ is computationally indistinguishable from $((g_1, \ldots, g_\ell, g_0), (g_1', \ldots, g_\ell', g_0'))$ (where g_i' are uniform). However, based on subgroup indistinguishability, a different method is required: Consider re-

[7] In this work, the interactive vector game is defined only for our subgroup indistinguishability assumptions, but it easily extends to other assumptions.

placing g_0 with $g_0' = h \cdot g_0$, the distribution $((g_1, \ldots, g_\ell, g_0), (g_1^r, \ldots, g_\ell^r, g_0^r))$ is computationally indistinguishable from $((g_1, \ldots, g_\ell, h \cdot g_0), (g_1^r, \ldots, g_\ell^r, h^r \cdot g_0^r))$ under the subgroup indistinguishability assumption. The crucial observation now is that since the orders of $\mathbb{G}_M$ and $\mathbb{G}_L$ are relatively prime, then in fact $g_0'^r = h^{r'} \cdot g_0^r$, where r' is independent of r. Combined with the fact that $\mathbb{G}_M$ is cyclic, we get that $((g_1, \ldots, g_\ell, g_0), (g_1^r, \ldots, g_\ell^r, g_0^r))$ is indistinguishable from $((g_1, \ldots, g_\ell, h \cdot g_0), (g_1^r \ldots g_\ell^r, h' \cdot g_0^r))$, for a random $h' \in \mathbb{G}_M$. Semantic security now follows.

To address the issues of circular security, leakage resiliency and auxiliary-input, we generalize the idea presented above, and prove that the distributions $((g_1, \ldots, g_\ell), (h^{a_1} \cdot g_1^r, \ldots, h^{a_\ell} \cdot g_\ell^r))$ and $((g_1, \ldots, g_\ell), (g_1^r, \ldots, g_\ell^r))$ are indistinguishable. We provide an interactive variant of this claim, which we call an *interactive ℓ-vector game*, where the values of $a_1, \ldots, a_\ell \in \mathbb{Z}$ are selected by the distinguisher and can depend on $(g_1, \ldots, g_\ell)$, and show that the above is hard even in such case. The interactive vector game will be employed in the proofs of all properties of the scheme.

For key-dependent message security, we consider the ciphertext $(g_0^r, g_1^r, \ldots, h \cdot g_i^r, \ldots, g_\ell^r)$. This ciphertext will be decrypted to h^{s_i} and in fact can be shown (using an interactive vector game) to be computationally indistinguishable from a legal encryption of h^{s_i}. Key-dependent message security follows from this fact.

Proving KDM$^{(n)}$-security for our scheme is more complex. To illustrate this, we contrast it with the ideas in the proof of [6]. They used homomorphism and re-randomization to achieve KDM$^{(n)}$-security: Their scheme is shown to have *homomorphic* properties that enable to "shift" public keys and ciphertexts that are relative to a certain secret key, into ones that are relative to another secret key. In order to apply these "shifts", one only needs to know the relation between the original and final keys (and not the keys themselves). In addition, their scheme is shown to have *re-randomization* properties that enable to take a public key (or ciphertext) and produce an independent public key (or ciphertext) that corresponds to the same secret key (and message, in the ciphertext case). These two properties enable simulating the KDM$^{(n)}$-security game using only one "real" secret key, fabricating the n required keys and ciphertexts using homomorphism and re-randomization. In [3], similar ideas are employed, but the re-randomization can be viewed as implicit in the assumption (the ability to generate independently looking vectors that are in fact linearly related).

Our scheme can be shown to have such homomorphic properties, but it doesn't enjoy as strong re-randomizability as required to use the above techniques. As an example, consider a public key $pk = (g_0, g_1, \ldots, g_\ell)$ corresponding to a secret key $sk = (s_1, \ldots, s_\ell)$, i.e. $g_0 = \prod g_i^{-s_i}$. Let $j \in [\ell]$ and consider $\hat{pk} = (\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_\ell)$ defined as follows: for all $i \notin \{j, 0\}$, set $\hat{g}_i = g_i$; for j, set $\hat{g}_j = g_j^{-1}$; and finally set $\hat{g}_0 = g_j \cdot g_0 = \hat{g}_j^{-(1-s_j)} \cdot \prod_{i \neq j} \hat{g}_i^{-s_i}$. We get that $\hat{pk}$ is a properly distributed public key corresponding to the secret key $\hat{sk} = sk \oplus e_j$ (sk XORed with the j^{th} unit binary string). Namely, we were able to "shift" a public key to correspond to another (related) secret key, without knowing the original key. However, the joint distribution of $pk, \hat{pk}$ is easily distinguishable from that

of two independent public keys. What we lack is the ability to re-randomize $\hat{pk}$ so that it is distributed as a public key for $\hat{sk}$ which is independent of pk.

Intuitively, this shortcoming requires us to use more "real randomness". Our proof simulates the KDM$^{(n)}$-security game using only one "real" secret key, as in the idea presented above. This secret key is used to fabricate n secret and public keys. However, when we want to apply the leftover hash lemma to claim that the g_0 components of all n fabricated public keys are close to uniform, we need the one real secret key to have sufficient entropy. This requires a secret key whose size is linear in n. These ideas, combined with the ones used to prove KDM$^{(1)}$ security, give our final proof.

The property of entropy-κ KDM-security requires that the scheme remains secure even when the secret key is sampled from a high-entropy (but not necessarily uniform) distribution. This is shown to hold using the leftover hash lemma, since $\prod g_i^{s_i}$ is a 2-universal hash function. A targeted encryption scheme is obtained similarly to the other constructions in [4], by using the fact that we can "fabricate" ciphertexts that correspond to affine functions of the secret key without knowing the secret key itself.

Leakage resiliency and auxiliary-input security are proven by an almost identical argument: consider a case where we replace the ciphertext $(h^m \cdot g_0^r, g_1^r, \ldots, g_\ell^r)$ with a computationally indistinguishable one: $(h^{-\sum \sigma_i s_i} \cdot h^m \cdot g_0^r, h^{\sigma_1} \cdot g_1^r, \ldots, h^{\sigma_\ell} \cdot g_\ell^r)$, where $\sigma_i \in \mathbb{Z}_M$ are uniform. Computational indistinguishability (even for a known secret key) follows from the interactive vector game mentioned above. For leakage-resilience, the leftover hash lemma implies that so long as there is sufficient entropy in $\mathbf{s}$ after the leakage, $\sum \sigma_i s_i$ will be close to uniform and will "mask" the value of m. For auxiliary input we use the *generalized Goldreich-Levin* theorem of [12] to show that $\sum \sigma_i s_i$ is close to uniform in the presence of a function of $\mathbf{s}$ that is hard to invert, even given the public key. Thus obtaining *weak* auxiliary-input security. In the QR case, the inner product is over $\mathbb{Z}_2$ and therefore we can use the "standard" Goldreich-Levin theorem [14], which implies better parameters. We use leveraging (as used in [12]) to obtain the full result.

1.4 Other Related Work

Cramer and Shoup [10] presented the notion of *hash proof systems*, which are similar to subgroup indistinguishability assumptions. Their implementations from QR and DCR also do not require the factorization of N in order to decrypt. However they use the discrete logarithm of (their analog to) the g_i's as a secret key for the system. Our scheme can be seen as taking another step towards "stripping" the secret key of all structure: in our scheme, it is just a uniform sequence of bits (resulting in a weaker form of a hash proof system that is "universal on average").

Hemenway and Ostrovsky [19] show how to construct lossy trapdoor functions (see [24] for definition) from the QR and DCR assumptions (among other assumptions). Similar ideas can be used in a straightforward manner to construct lossy trapdoor functions from subgroup indistinguishability assumptions with special properties.

1.5 Paper Organization

Due to space constraints, this extended abstract only discusses the construction based on the QR assumption. In addition, some of the proofs are omitted. We refer the reader to the full version of this paper [7] for the complete presentation, including all details.

Preliminaries and definitions are presented in Section 2. The definition of subgroup indistinguishability assumptions and instantiations from QR and DCR appear in Section 3.

Our QR-based encryption scheme is presented in Section 4, followed, in Section 5, by introduction of the interactive vector game: a central technical tool to be used for the analysis throughout the paper. KDM-security is discussed in Section 6, leakage-resilience in Section 7 and auxiliary-input security in Section 8.

2 Preliminaries

We denote scalars in plain lowercase ($x \in \{0,1\}$) and vectors in bold lowercase ($\mathbf{x} \in \{0,1\}^n$). The i^{th} coordinate of $\mathbf{x}$ is denoted x_i.

For vectors $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$, where $\mathbb{G}$ is a multiplicative commutative group, we denote by $\mathbf{g}^r$ the vector whose i^{th} coordinate is g_i^r. We denote by $\mathbf{h} \cdot \mathbf{g}$ the vector whose i^{th} coordinate is $h_i \cdot g_i$. Note that this does *not* denote an inner product. For a group element $g \in \mathbb{G}$ and a vector $\mathbf{x} \in \mathbb{Z}$, we let $g^{\mathbf{x}}$ denote the vector whose i^{th} coordinate is g^{x_i}.

Let X be a probability distribution over a domain S, we write $x \xleftarrow{\$} X$ to indicate that x is sampled from the distribution X. The uniform distribution over a set S is denoted $U(S)$. We use $x \xleftarrow{\$} S$ as abbreviation for $x \xleftarrow{\$} U(S)$. An ensemble $X = \{X_k\}_k$ is $\epsilon = \epsilon(k)$-*uniform* if for all k, X_k is within statistical distance $\epsilon(k)$ from the uniform distribution. Statistical and computational indistinguishability are defined in the standard way. We write $\text{negl}(k)$ to denote an arbitrary *negligible* function, i.e. one that vanishes faster than the inverse of any polynomial.

We use the following simple lemma.

Lemma 2.1. *Let $T, N \in \mathbb{N}$ and let $x \xleftarrow{\$} [T]$, then $x \pmod N$ is (N/T)-uniform in $\mathbb{Z}_N$.*

We use the following lemma which is an immediate corollary of the leftover hash lemma and explicitly appears in [6, Lemma 2].

Lemma 2.2. *Let H be a 2-universal hash family from a set X to a set Y. Then the distribution $(h, h(x))$ where $h \xleftarrow{\$} H$, $x \xleftarrow{\$} X$ is $\sqrt{\frac{|Y|}{4|X|}}$-uniform in $H \times Y$.*

The following lemma states the properties of a class of hash functions that we use.

Lemma 2.3. *Let $\mathbb{G}$ be any finite commutative group and let $\ell \in \mathbb{N}$. Then the set of functions $H = \{h_{g_1,\ldots,g_\ell} : \{0,1\}^\ell \to \mathbb{G}\}_{g_1,\ldots,g_\ell \in \mathbb{G}}$ where $h_{g_1,\ldots,g_\ell}(\mathbf{x}) = \prod_{i \in [\ell]} g_i^{x_i}$, is 2-universal.*

We use the standard definitions of KDM security, leakage resilience and auxiliary input security as appear, e.g., in [6, 22, 12], respectively.

3 Subgroup Indistinguishability Assumptions

We present the class of subgroup indistinguishability assumptions in Section 3.1 and then discuss instantiations under the QR and DCR assumptions in Section 3.2.

3.1 Definition of a Subgroup Indistinguishability (SG) Problem

Let $\mathbb{G}_U$ be a finite commutative multiplicative group, such that $\mathbb{G}_U$ is a direct product of two groups: $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ (interpreted as the "message group" and the "language group"), where $\mathbb{G}_M$ is cyclic of order M, $\mathbb{G}_L$ is of order L (and is not necessarily cyclic) and $\mathbb{G}_U$ is of order $M \cdot L$ (we abuse notation and use M, L to index the groups and to denote their orders). We require that $\gcd(M, L) = 1$. Let h be a generator for $\mathbb{G}_M$ such that h is efficiently computable from the description of $\mathbb{G}_U$. We require that there exists an efficient algorithm $OP_{\mathbb{G}_U}$ to perform group operations in $\mathbb{G}_U$, and also that there exist efficient sampling algorithms $S_{\mathbb{G}_M}$, $S_{\mathbb{G}_L}$ that sample a random element from $\mathbb{G}_M$, $\mathbb{G}_L$ respectively. We further require that an upper bound $T \geq M \cdot L$ is known.

We stress that as always, all groups described above are in fact families of groups, indexed by the security parameter k. To be more precise, there exists a polynomial time randomized algorithm that given the security parameter 1^k, outputs $I_{\mathbb{G}_U} = (OP_{\mathbb{G}_U}, S_{\mathbb{G}_M}, S_{\mathbb{G}_L}, h, T)$. We refer to $I_{\mathbb{G}_U}$ as *an instance* of $\mathbb{G}_U$.

For any adversary $\mathcal{A}$ we denote the *subgroup distinguishing advantage* of $\mathcal{A}$ by

$$\mathrm{SGAdv}[\mathcal{A}] = \left| \Pr_{x \xleftarrow{\$} \mathbb{G}_U} [\mathcal{A}(1^k, x)] - \Pr_{x \xleftarrow{\$} \mathbb{G}_L} [\mathcal{A}(1^k, x)] \right| .$$

That is, the advantage $\mathcal{A}$ has in distinguishing between $\mathbb{G}_U$ and $\mathbb{G}_L$. The *subgroup indistinguishability* (SG) assumption is that for any polynomial $\mathcal{A}$ it holds that for a properly sampled instance $I_{\mathbb{G}_U}$, we have $\mathrm{SGAdv}[\mathcal{A}] = \mathrm{negl}(k)$ (note that in such case it must be that $1/L = \mathrm{negl}(k)$). In other words, thinking of $\mathbb{G}_L \subseteq \mathbb{G}_U$ as a language, the assumption is that this language is hard on average. We define an additional flavor of the assumption by

$$\mathrm{SG'Adv}[\mathcal{A}] = \left| \Pr_{x \xleftarrow{\$} \mathbb{G}_L} [\mathcal{A}(1^k, h \cdot x)] - \Pr_{x \xleftarrow{\$} \mathbb{G}_L} [\mathcal{A}(1^k, x)] \right| .$$

It follows immediately that for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{B}$ such that $\mathrm{SG'Adv}[\mathcal{A}] \leq 2 \cdot \mathrm{SGAdv}[\mathcal{B}]$.

3.2 Instantiations

We instantiate the SG assumption based on the QR and DCR assumptions.

For both instantiations we consider a modulus N defined as follows. For security parameter k, we sample a random *RSA number* $N \in \mathbb{N}$: this is a number of the form $N = pq$ where p, q are random k-bit odd primes.

We note that our instantiations work even when the modulus N is such that $\mathbb{QR}_N$ is not cyclic.

Instantiation Under the QR Assumption with Any Blum Integer Consider a modulus N as described above. We use $\mathbb{J}_N$ to denote the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol 1, we use $\mathbb{QR}_N$ to denote the set of *quadratic residues* (squares) modulo N. Slightly abusing notation $\mathbb{J}_N, \mathbb{QR}_N$ also denote the respective groups with the multiplication operation modulo N. The groups $\mathbb{J}_N, \mathbb{QR}_N$ have orders $\frac{\varphi(N)}{2}, \frac{\varphi(N)}{4}$ respectively and we denote $N' = \frac{\varphi(N)}{4}$. We require that N is a *Blum integer*, namely that $p, q = 3 \pmod 4$. In such case it holds that $\gcd(2, N') = 1$ and $(-1) \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

The *quadratic residuosity* (QR) assumption is that for a properly generated N, the distributions $U(\mathbb{J}_N)$ and $U(\mathbb{QR}_N)$ are computationally indistinguishable.[8] This leads to the immediate instantiation of the SG assumption by setting $\mathbb{G}_U = \mathbb{J}_N$, $\mathbb{G}_M = \{\pm 1\}$, $\mathbb{G}_L = \mathbb{QR}_N$, $h = (-1)$, $T = N \geq 2N'$.

Instantiation Under the DCR Assumption The *decisional composite residuosity* (DCR) assumption, introduced by Paillier [23], states that for a properly generated RSA number N, it is hard to distinguish between a random element in $\mathbb{Z}_{N^2}^*$ and a random element in the subgroup of N^{th}-residues $\{x^N : x \in \mathbb{Z}_{N^2}^*\}$. The group $\mathbb{Z}_{N^2}^*$ can be written as a product of the group generated by $1 + N$ (which has order N) and the group of N^{th} residues (which has order $\varphi(N)$). This implies that setting $\mathbb{G}_U = \mathbb{Z}_{N^2}^*$, $\mathbb{G}_L = \{x^N : x \in \mathbb{Z}_{N^2}^*\}$ and $\mathbb{G}_M = \{(1+N)^i : i \in [N]\}$ provides an instantiation of the SG assumption, setting $h = (1+N)$ and $T = N^2$. It is left to check that indeed $\gcd(N, \varphi(N)) = 1$. This follows since p, q are odd primes of equal length: assume w.l.o.g that $p/2 < q < p$, then the largest prime divisor of $\varphi(N) = (p-1)(q-1)$ has size at most $(p-1)/2 < p, q$ and the claim follows.[9]

4 Description of the Encryption Scheme

We now present our QR-based scheme $\mathcal{E}[\ell]$.

Parameters. The scheme is parameterized by $\ell \in \mathbb{N}$ which is polynomial in the security parameter. The exact value of ℓ is determined based on the specific properties we require from the scheme.

[8] The QR assumption usually refers to random RSA numbers, which are not necessarily Blum integers. However, since Blum integers have constant density among RSA numbers, the flavor we use is implied.

[9] If greater efficiency is desired, we can use a generalized form of the assumption, presented in [11].

The message space of $\mathcal{E}[\ell]$ is $\mathcal{M} = \{0,1\}$, i.e. this is a bit-by-bit encryption scheme.

Key generation. The key generator first samples a Blum integer N. We note that the same value of N can be used by all users. Furthermore we stress that no entity needs to know the factorization of N. Therefore we often refer to N as a public parameter of the scheme and assume that it is implicitly known to all users.

The key generator also samples $\mathbf{s} \xleftarrow{\$} \{0,1\}^\ell$ and sets $sk = \mathbf{s}$. It then samples $\mathbf{g} \xleftarrow{\$} \mathbb{QR}_N^\ell$ and sets $g_0 = (\prod_{i\in[\ell]} g_i^{s_i})^{-1}$. The public key is set to be $pk = (g_0, \mathbf{g})$ (with N as an additional implicit public parameter).

Encryption. On inputs a public key $pk = (g_0, \mathbf{g})$ and a message $m \in \{0,1\}$, the encryption algorithm runs as follows: it samples $r \xleftarrow{\$} [N^2]$,[10] and computes $\mathbf{c} = \mathbf{g}^r$ and $c_0 = (-1)^m \cdot g_0^r$. It outputs a ciphertext $(c_0, \mathbf{c})$.

Decryption. On inputs a secret key $sk = \mathbf{s}$ and a ciphertext $(c_0, \mathbf{c})$, the decryption algorithm computes $(-1)^m = c_0 \cdot \prod_{i\in[\ell]} c_i^{s_i}$ and outputs m.

The completeness of the scheme follows immediately by definition.

5 The Interactive Vector Game

We define the *interactive ℓ-vector* game played between a challenger and an adversary. We only present the QR-based game and refer the reader to [7] for full details.

Initialize. The challenger samples $b \xleftarrow{\$} \{0,1\}$ and also generates a Blum integer N and a vector $\mathbf{g} \xleftarrow{\$} \mathbb{QR}_N^\ell$. It sends N and $\mathbf{g}$ to the adversary.

Query. The adversary adaptively makes queries, where each query is a vector $\mathbf{a} \in \{0,1\}^\ell$. For each query $\mathbf{a}$, the challenger samples $r \xleftarrow{\$} [N^2]$ and returns $(-1)^{\mathbf{a}} \cdot \mathbf{g}^r$ if $b = 0$ and $\mathbf{g}^r$ if $b = 1$.

Finish. The adversary outputs a guess $b' \in \{0,1\}$.

The advantage of an adversary $\mathcal{A}$ in the game is defined to be

$$\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}] = |\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| \ .$$

Under the QR assumption, no poly(k)-time adversary (where k is the security parameter) can obtain a non-negligible advantage in the game, as formally stated below.

Lemma 5.1. *Let $\mathcal{A}$ be an adversary for the interactive ℓ-vector game that makes at most t queries, then there exists an adversary $\mathcal{B}$ for* QR *such that*

$$\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}] \leq 4t\ell \cdot \mathrm{QRAdv}[\mathcal{B}] + 2t\ell/N \ .$$

[10] A more natural choice is to sample $r \xleftarrow{\$} [|\mathbb{J}_N|]$, but since $|\mathbb{J}_N| = 2N' = \frac{\varphi(N)}{2}$ is hard to compute, we cannot sample from this distribution directly. However, since r is used as an exponent of a group element, it is sufficient that $(r \bmod 2N')$ is uniform in $\mathbb{Z}_{2N'}$, and this is achieved by sampling r from a much larger domain. We further remark that for the QR case, it is in fact sufficient to use $r \xleftarrow{\$} [(N-3)/4]$.

Proof. A standard hybrid argument implies the existence of $\mathcal{A}_1$ which is an adversary for a 1-round game ($t = 1$ in our notation) such that $\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}] \leq t \cdot \mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}_1]$.

We consider a series of hybrids (experiments). For each hybrid H_i, we let $\Pr[H_i]$ denote the probability that the experiment "succeeds" (an event we define below).

Hybrid H_0. In this experiment, we flip a coin $b \xleftarrow{\$} \{0,1\}$ and also sample $i \xleftarrow{\$} [\ell]$. We simulate the 1-round game with $\mathcal{A}_1$ where the challenger answers a query $\mathbf{a}$ with $(g_1^r, \ldots, g_{i-1}^r, (-1)^{b \cdot a_i} \cdot g_i^r, (-1)^{a_{i+1}} \cdot g_{i+1}^r, \ldots, (-1)^{a_\ell} \cdot g_\ell^r)$. The experiment succeeds if $b' = b$.

A standard argument shows that

$$\frac{\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}_1]}{2\ell} = \left| \Pr[H_0] - \frac{1}{2} \right| .$$

Hybrid H_1. In this hybrid we replace g_i (which is a uniform square) with $(-g_i)$. We get that there exists $\mathcal{B}$ such that $|\Pr[H_1] - \Pr[H_0]| \leq 2 \cdot \mathrm{QRAdv}[\mathcal{B}]$.

We note that in this hybrid the adversary's query is answered with

$$(g_1^r, \ldots, g_{i-1}^r, (-1)^{b \cdot a_i} \cdot (-g_i)^r, (-1)^{a_{i+1}} \cdot g_{i+1}^r, \ldots, (-1)^{a_\ell} \cdot g_\ell^r) .$$

Hybrid H_2. In this hybrid the only change is that now $r \xleftarrow{\$} \mathbb{Z}_{2N'}$ (recall that $N' = \frac{\varphi(N)}{4}$) rather than $U([N^2])$. By Lemma 2.1 it follows that $|\Pr[H_2] - \Pr[H_1]| \leq 1/N$. We note that while N' is not explicitly known to any entity, this argument is statistical and there is no requirement that this hybrid is efficiently simulated.

We denote $r_1 = (r \bmod 2)$ and $r_2 = (r \bmod N')$. Since N' is odd, the Chinese Remainder Theorem implies that r_1, r_2 are uniform in $\mathbb{Z}_2, \mathbb{Z}_{N'}$ respectively and are independent. The answer to the query in this scenario is therefore

$$\begin{aligned}(g_1^r, \ldots, g_{i-1}^r, (-1)^{b \cdot a_i} \cdot (-g_i)^r, (-1)^{a_{i+1}} \cdot g_{i+1}^r, \ldots, (-1)^{a_\ell} \cdot g_\ell^r) = \\ (g_1^{r_2}, \ldots, g_{i-1}^{r_2}, (-1)^{b \cdot a_i + r_1} \cdot g_i^{r_2}, (-1)^{a_{i+1}} \cdot g_{i+1}^{r_2}, \ldots, (-1)^{a_\ell} \cdot g_\ell^{r_2}) .\end{aligned}$$

However since r_1 is a uniform bit, the answer is independent of b. It follows that $\Pr[H_2] = \frac{1}{2}$. Thus $\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}_1] \leq 4\ell \cdot \mathrm{QRAdv}[\mathcal{B}] + 2\ell/N$, and the result follows. □

6 KDM Security

In this section, we discuss the KDM-security related properties of our QR-based scheme (for the general discussion and full details, see full version [7]). We prove the $\mathrm{KDM}^{(1)}$-security of $\mathcal{E}[\ell]$, for $\ell \geq \log N + \omega(\log k)$, in Section 6.1. Then, in Section 6.2, we state that for $\ell \geq n \cdot \log N + \omega(\log k)$, $\mathcal{E}[\ell]$ is also $\mathrm{KDM}^{(n)}$-secure. Finally, extensions beyond affine functions are stated in Section 6.3.

We define $\mathcal{F}_{\mathrm{aff}}$ to be the class of affine functions over $\mathbb{Z}_2$. Namely, all functions of the form $f_{a_0,\mathbf{a}}(\mathbf{x}) = a_0 + \sum a_i x_i$, where $a_i, x_i \in \mathbb{Z}_2$.

We use $\mathrm{KDM}_{\mathcal{F}}\mathrm{Adv}[\mathcal{A}]$ to denote the advantage of an adversary $\mathcal{A}$ in distinguishing between a case where it gets legal encryptions of functions in $\mathcal{F}$ and the case where it gets encryptions of the constant message 0.

6.1 KDM$^{(1)}$-Security

The intuition behind the KDM$^{(1)}$-security of $\mathcal{E}[\ell]$ is as follows. Consider a public key $(g_0 = \prod g_i^{-s_i}, \mathbf{g})$ that corresponds to a secret key $\mathbf{s}$, and a function $f_{a_0,\mathbf{a}} \in \mathcal{F}_{\text{aff}}$. The encryption of $f_{a_0,\mathbf{a}}(\mathbf{s}) = (-1)^{a_0+\sum a_i s_i}$ is

$$(c_0, \mathbf{c}) = ((-1)^{a_0+\sum a_i s_i} \cdot g_0^r, \mathbf{g}^r) = ((-1)^{a_0} \cdot \prod((-1)^{a_i} \cdot g_i^r)^{-s_i}, \mathbf{g}^r) \ .$$

We notice that if $\mathbf{s}, a_0, \mathbf{a}$ are known, then c_0 is completely determined by $\mathbf{c} = \mathbf{g}^r$. Therefore, if we replace $\mathbf{g}^r$ with $(-1)^{\mathbf{a}} \cdot \mathbf{g}^r$ (an indistinguishable vector, even given the public key, by an interactive vector game), we see that $(c_0, \mathbf{c})$ is indistinguishable from $(c_0', \mathbf{c}') = ((-1)^{a_0} \cdot g_0^r, (-1)^{\mathbf{a}} \cdot \mathbf{g}^r)$, even when the secret key and the message are known. Applying the same argument again, taking into account that g_0 is close to uniform, implies that $(c_0', \mathbf{c}')$ is computationally indistinguishable from $(g_0^r, \mathbf{g}^r)$, which is an encryption of 0. A formal statement and analysis follow.

Theorem 6.1. *Let $\mathcal{A}$ be a $\text{KDM}^{(1)}_{\mathcal{F}_{\text{aff}}}$-adversary for $\mathcal{E}[\ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that*

$$\text{KDM}^{(1)}_{\mathcal{F}_{\text{aff}}}\text{Adv}[\mathcal{A}] \leq 4t(2\ell+1) \cdot \text{QRAdv}[\mathcal{B}] + \sqrt{N \cdot 2^{-\ell}} + O(t\ell/N) \ .$$

The theorem implies that taking $\ell = \log N + \omega(\log k)$ is sufficient to obtain KDM$^{(1)}$-security.

Proof. The proof proceeds by a series of hybrids. Let b' denote $\mathcal{A}$'s output.

Hybrid H_0. In this hybrid, the adversary gets the public key, queries functions $f_{a_0,\mathbf{a}} \in \mathcal{F}_{\text{aff}}$ and gets legal encryptions of the functions of the secret key.

Hybrid H_1. In this hybrid, we change the way the challenger answers the adversary's queries. Recall that in hybrid H_0, the query $f_{a_0,\mathbf{a}} \in \mathcal{F}_{\text{aff}}$ was answered by $(c_0, \mathbf{c}) = ((-1)^{a_0+\sum a_i s_i} \cdot g_0^r, \mathbf{g}^r)$. In hybrid H_1, it will be answered by $(c_0, \mathbf{c}) = ((-1)^{a_0} \cdot g_0^r, (-1)^{\mathbf{a}} \cdot \mathbf{g}^r)$.

We prove that

$$\left|\Pr_{H_1}[b'=1] - \Pr_{H_0}[b'=1]\right| \leq \text{IV}_\ell\text{Adv}[\mathcal{A}'] \leq 4t\ell \cdot \text{QRAdv}[\mathcal{B}_1] + O(t\ell/N) \ ,$$

for some $\mathcal{A}', \mathcal{B}_1$, even when $\mathbf{s}$ is fixed and known.

To see this, we notice that in both hybrids $c_0 = (-1)^{a_0} \cdot \prod_{i\in[\ell]}((-1)^{a_i} \cdot c_i^{-1})^{s_i}$ and $g_0 = \prod_{i\in[\ell]} g_i^{-s_i}$. Therefore an adversary $\mathcal{A}'$ for the interactive ℓ-vector game can simulate $\mathcal{A}$, sampling $\mathbf{s}$ on its own and using $\mathbf{g}$ to generate g_0 and "translate" the challenger answers. Applying Lemma 5.1, the result follows.

Hybrid H_2. In this hybrid, we change the distribution of g_0, which will now be sampled from $U(\mathbb{QR}_N)$. By Lemma 2.3 combined with Lemma 2.2, $(g_0, \mathbf{g})$ is $\sqrt{\frac{N'}{2^{\ell+2}}} \leq \sqrt{\frac{N}{2^{\ell+2}}}$-uniform. Thus

$$\left|\Pr_{H_2}[b'=1] - \Pr_{H_1}[b'=1]\right| \leq \sqrt{\frac{N}{2^{\ell+2}}} \ .$$

Hybrid H_3. In this hybrid, we again change the way the challenger answers queries. Now instead of answering $(c_0, \mathbf{c}) = ((-1)^{a_0} \cdot g_0^r, (-1)^{\mathbf{a}} \cdot \mathbf{g}^r))$, the challenger answers $(c_0, \mathbf{c}) = (g_0^r, \mathbf{g}^r)$. The difference between H_2 and H_3 is now a t-query interactive $(\ell+1)$-vector game and thus by Lemma 5.1,

$$\left| \Pr_{H_3}[b' = 1] - \Pr_{H_2}[b' = 1] \right| \leq 4t(\ell+1) \cdot \mathrm{QRAdv}[\mathcal{B}_2] + O(t\ell/N) \ ,$$

for some $\mathcal{B}_2$.

Hybrid H_4. We now revert the distribution of g_0 back to the original $\prod_{i \in [\ell]} g_i^{-s_i}$. Similarly to H_2, we have

$$\left| \Pr_{H_4}[b' = 1] - \Pr_{H_3}[b' = 1] \right| \leq \sqrt{\frac{N}{2^{\ell+2}}} \ .$$

However, hybrid H_4 is identical to answering all the queries of the adversary by encryptions of 0. Summing the terms above, the result follows. □

6.2 KDM$^{(n)}$-Security

A formal statement for the QR case follows.

Theorem 6.2. *Let $\mathcal{A}$ be a $\mathrm{KDM}^{(n)}_{\mathcal{F}_{\mathrm{aff}}}$-adversary for $\mathcal{E}[\ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that*

$$\mathrm{KDM}^{(n)}_{\mathcal{F}_{\mathrm{aff}}}\mathrm{Adv}[\mathcal{A}] \leq 4nt(2\ell+1) \cdot \mathrm{QRAdv}[\mathcal{B}] + (N \cdot 2^{-\ell/n})^{n/2} + O(nt\ell/N) \ .$$

Thus, taking $\ell = n \cdot \log N + \omega(\log k)$ is sufficient for KDM$^{(n)}$-security.

6.3 Beyond Affine Functions

Two building blocks have been suggested in [8, 4] to obtain KDM-security w.r.t. a larger class of functions. Our scheme has the properties required to apply both constructions, yielding the following corollaries (that can be generalized to any SG assumption, see full version [7]).

The first corollary is derived using [8, Theorem 1.1]. A set of functions $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i : \{0,1\}^\kappa \to \{0,1\}\}$ is *entropy preserving* if the function $f(x) = (h_1(x) \| \cdots \| h_\ell(x))$ is injective (the operator $\|$ represents string concatenation).

Corollary 6.1. *Consider $\mathcal{E}[\ell]$ and let κ be polynomial in the security parameter such that $\kappa \geq \log N + \omega(\log k)$. Then for any entropy preserving set $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i \in \{0,1\}^\kappa \to \{0,1\}\}$ of efficiently computable functions, with polynomial cardinality (in the security parameter), there exists a* KDM$^{(1)}$*-secure scheme under the* QR*-assumption w.r.t. the class of functions*

$$\mathcal{F} = \left\{ f(\mathbf{x}) = a_0 + \sum a_i h_i(\mathbf{x}) : (a_0, \mathbf{a}) \in \mathbb{Z}_2 \times \mathbb{Z}_2^\ell \right\} \ .$$

The second corollary is derived using [4, Theorem 4.1].

Corollary 6.2. *Based on the* QR *assumption, for any polynomial p there exists a* $\mathrm{KDM}^{(1)}$*-secure encryption scheme w.r.t. all functions computable by circuits of size $p(k)$ (where k is the security parameter).*

7 Leakage Resiliency

We prove that the scheme $\mathcal{E}[\ell]$ (our QR based scheme) is resilient to a leakage of up of $\lambda = \ell - \log N - \omega(\log k)$ bits. This implies that taking $\ell = \omega(\log N)$, achieves $(1 - o(1))$ leakage rate.

Intuitively, to prove leakage resiliency, we consider the case where instead of outputting the challenge ciphertext $((-1)^m \cdot g_0^r, \mathbf{g}^r)$, we output $((-1)^m \cdot (-1)^{\sum \sigma_i s_i} \cdot g_0^r, (-1)^{\boldsymbol{\sigma}} \cdot \mathbf{g}^r)$, for a random vector $\boldsymbol{\sigma} \stackrel{\$}{\leftarrow} \mathbb{Z}_2^\ell$. The views of the adversary in the two cases are indistinguishable (by an interactive vector game).[11] Using the leftover hash lemma, so long as $\mathbf{s}$ has sufficient min-entropy, even given g_0 and the leakage, then $\sum \sigma_i s_i$ is close to uniform. In other words, the ciphertexts generated by our scheme are computationally indistinguishable from ones that contain a strong extractor (whose seed is the aforementioned $\boldsymbol{\sigma}$), applied to the secret key. This guarantees leakage resiliency.[12] The result in the QR case is formally stated below, where $\mathrm{Leak}_\lambda\mathrm{Adv}[\mathcal{A}]$ denotes the advantage of an adversary $\mathcal{A}$ in breaking the security of the scheme using λ bits of leakage.

Theorem 7.1. *Let $\mathcal{A}$ be a λ-leakage adversary for $\mathcal{E}[\ell]$. Then there exists an adversary $\mathcal{B}$ such that*

$$\mathrm{Leak}_\lambda\mathrm{Adv}[\mathcal{A}] \leq 8\ell \cdot \mathrm{QRAdv}[\mathcal{B}] + \sqrt{N \cdot 2^{\lambda-\ell}} + O(\ell/N) \ .$$

8 Auxiliary-Input Resiliency

As in previous work, we start by stating weak auxiliary-input security in Lemma 8.1 below and then derive general auxiliary-input security for sub-exponentially hard functions in Corollary 8.1.

A function f is ϵ-weakly uninvertible if for any efficient $\mathcal{A}$, $\Pr[\mathcal{A}(1^k, pk, f_k(sk, pk)) = sk] \leq \epsilon(|sk|)$.

Lemma 8.1. *Let $\epsilon(\ell)$ and f be such that ϵ is negligible and f is ϵ-weakly uninvertible function (more precisely, family of functions). Then under the* QR *assumption, the scheme $\mathcal{E}[\ell]$ is secure even with auxiliary input $f(sk)$.*

[11] Of course the latter ciphertext can only be generated using the secret key, but the indistinguishability holds even when the secret key is known.

[12] In the spirit of [22], we can say that our scheme defines a new hash proof system that is universal with high probability over illegal ciphertexts, a property which is sufficient for leakage resiliency.

We note that the above may seem confusing since it appears to imply auxiliary-input security, and thus also semantic security, regardless of the value of ℓ. However, we recall that if ℓ is too small, then we may be able to retrieve $\mathbf{s}$ from pk without the presence of any auxiliary input. Therefore the value of ℓ must be large enough in order for f to be weakly uninvertible.

We can then derive the following corollary.

Corollary 8.1. *Assuming that a subgroup indistinguishability assumption holds, then for any constant $\delta > 0$ there is an encryption scheme that is resilient to auxiliary input $f(sk)$ any function f is hard to invert with probability $2^{-\ell^\delta}$.*

Acknowledgments The authors wish to thank Gil Segev for illuminating discussions. The first author wishes to thank Microsoft Research New-England, for hosting him at the time of this research.

References

1. Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 374–396. Springer, 2005.
2. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.
3. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Halevi [18], pages 595–618.
4. Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *EUROCRYPT*, 2010. To appear.
5. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 2002.
6. Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.
7. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: Quadratic residuosity strikes back). Cryptology ePrint Archive, Report 2010/226, 2010. `http://eprint.iacr.org/`.
8. Zvika Brakerski, Shafi Goldwasser, and Yael Kalai. Circular-secure encryption beyond affine functions. Cryptology ePrint Archive, Report 2009/485, 2009. `http://eprint.iacr.org/`.
9. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

10. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.
11. Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer, 2001.
12. Yevgeniy Dodis, Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In TCC 2010 (to appear), 2010.
13. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *STOC*, pages 621–630. ACM, 2009.
14. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. ACM, 1989.
15. Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377. ACM, 1982.
16. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
17. J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.
18. Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.
19. Brett Hemenway and Rafail Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16(127), 2009. `http://eccc.uni-trier.de/report/2009/127/`.
20. Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 590–609. Springer, 2009.
21. Peeter Laud and Ricardo Corin. Sound computational interpretation of formal encryption with composed keys. In Jong In Lim and Dong Hoon Lee, editors, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 55–66. Springer, 2003.
22. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [18], pages 18–35.
23. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.
24. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 187–196. ACM, 2008.
25. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 84–93. ACM, 2005.