Circular and Leakage Resilient Public-Key Encryption Under Subgroup Indistinguishability

(or: Quadratic Residuosity Strikes Back)

Zvika Brakerski* Shafi Goldwasser†

May 25, 2010

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 [BGK, ePrint09] or [BHHI, Eurocrypt10].
- **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 [NS, Crypto09], 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.

*Weizmann Institute of Science, zvika.brakerski@weizmann.ac.il.

†Weizmann Institute of Science and Massachusetts Institute of Technology, shafi@theory.csail.mit.edu.

1 Introduction

The "classical" definition of *semantic secure* public-key encryption by Goldwasser and Micali [GM84], 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 [GM84] 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 previous 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 [GM84]. It was later observed that this situation is not so unlikely and may sometimes even be desirable [CL01, ABHS05, LC03]. Black, Rogoway and Shrimpton [BRS02] 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 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 $\text{KDM}^{(n)}$-security while the single-user case is called $\text{KDM}^{(1)}$-security.

Boneh, Halevi, Hamburg and Ostrovsky [BHHO08] constructed a public key encryption scheme that is $\text{KDM}^{(n)}$ secure w.r.t. all affine functions,[1] 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 [ACPS09] who proved that a variation of Regev's scheme [Reg05] 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 [BGK09] and by Barak, Haitner, Hofheinz and Ishai [BHHI09] presented each general and different techniques to extend KDM-security to richer classes of functions. In [BGK09], 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 [BHHI09], 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 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. [HSH+08], gave rise to the notion of public-key encryption resilient to (bounded) memory leakage attacks, presented by Akavia, Goldwasser and Vaikuntanathan [AGV09] and further explored by Naor and Segev [NS09]. 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.[2] [AGV09, NS09] presented public-key encryption schemes that are resilient to leakage of even a $1 - o(1)$ fraction of the secret key (we

[1] 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 $h \cdot \prod g_i^{a_i}$.

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

call this the "leakage rate"). In particular, [AGV09] showed how this can be achieved under the LWE assumption, while [NS09] showed that this can be achieved under the DDH (or d-linear) assumption. It is further shown in [NS09] that some leakage resilience can be achieved using any universal hash proof system (defined in [CS02]), 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 [NS09, CS02, DJ01].

Auxiliary input. Dodis, Kalai and Lovett [DKL09] and Dodis, Goldwasser, Kalai, Peikert and Vaikuntanathan [DGK+10] 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 [DGK+10]). 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 [DGK+10].

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 $\mathbb{G}_L$ 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,[3] 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,[4] 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.

We remark that the basic structure of our construction is strikingly similar to [BHHO08], 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 [BHHO08], 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 [ACPS09]) 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 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 KDM$^{(1)}$ with respect to the affine functions "in the exponent": for any

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

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

$h_0, h_1, \ldots, h_\ell \in \mathbb{G}_M$ 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 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 KDM$^{(n)}$ security. Interestingly, in the works of [BHHO08, ACPS09], 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 [BGK09] 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 [BHHI09], 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 [AGV09] or DDH [NS09], and even (implicitly) under DCR [NS09, CS02], this was not the case under QR. Previous results with regards to QR-based leakage resiliency [NS09, CS02] 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 [DGK$^+$10]. In turn, [DGK$^+$10] 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 [BHHO08, ACPS09, NS09, DGK$^+$10].[5]

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 [GM82]). 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 [BHHO08]. 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 replacing g_0 with $g_0' =$

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

$h \cdot g_0$, 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 distribution $((g_1, \ldots, g_\ell), (h^{a_1} \cdot g_1^r, \ldots, h^{a_\ell} \cdot g_\ell^r))$ is indistinguishable from $((g_1, \ldots, g_\ell), (g_1^r, \ldots, g_\ell^r))$. 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 [BHHO08]. 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 [ACPS09], 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 the original 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 [BHHI09], 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 [DGK+10] 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 [GL89], which implies better parameters. We use leveraging (as used in [DGK+10]) to obtain the full result.

1.4 Other Related Work

Cramer and Shoup [CS02] 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 [HO09] show how to construct lossy trapdoor functions (see [PW08] 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

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

In the interest of clarity and to simplify our presentation, the body of the paper only discusses the construction based on the QR assumption. Some of the proofs are omitted from this part. The general case, for any subgroup indistinguishability assumption, is presented and analyzed in detail in the appendix.

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.

In Appendix A we present a variant of [DGK+10, Theorem 1], together with a proof, for the sake of completeness. Appendix B contains the general presentation of our construction from subgroup

indistinguishability assumptions, together with the full proofs.

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. The vector inner product of $\mathbf{x}, \mathbf{y}$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$ and is defined to be $\sum x_i \cdot y_i$. The vector $\mathbf{e_i}$ is the i^{th} unit vector.

For any $K \in \mathbb{N}$, we denote $[K] = \{1, \ldots, K\}$. We let $\mathbb{Z}_K$ denote the ring $\mathbb{Z}/K\mathbb{Z}$ and let $\mathbb{Z}_K^*$ denote the group of units in $\mathbb{Z}_K$. Euler's totient function is denoted by $\varphi(\cdot)$.

Arithmetic operations are always performed in $\mathbb{Z}$ but can sometimes also be interpreted as being performed over $\mathbb{Z}_K$, for some value $K \in \mathbb{N}$. Specifically, if h is an element of order K in a multiplicative group, then $h^x = h^{(x \bmod K)}$, so operations "in the exponent of h" can be interpreted as operations over $\mathbb{Z}_K$. We usually write (mod K) to indicate that an operation is performed over $\mathbb{Z}_K$, but we sometimes omit this when K is clear from the context.

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 $x \xleftarrow{\$} U(S)$. For any function f with domain S we let $f(X)$ denote the random variable (or corresponding distribution) obtained by sampling $x \xleftarrow{\$} X$ and outputting $f(x)$. The *min-entropy* of a (discrete) random variable X is $\mathbf{H}_\infty(X) = \min_{x \in S}\{-\log \Pr[X = x]\}$.

We write $\text{negl}(k)$ to denote an arbitrary *negligible* function, i.e. one that vanishes faster than the inverse of any polynomial.

The *statistical distance* between two distributions X, Y (or random variables with those distributions) over a common domain S is $\max_{A \subseteq S} |\Pr[X \in A] - \Pr[Y \in A]|$. Two ensembles $X = \{X_k\}_k$, $Y = \{Y_k\}_k$ are $\epsilon = \epsilon(k)$-*close* if for all k, the distance between X_k and Y_k is at most $\epsilon(k)$ and are *statistically indistinguishable* if $\epsilon(k) = \text{negl}(k)$. An ensemble $X = \{X_k\}_k$ over domains $S = \{S_k\}_k$ is $\epsilon = \epsilon(k)$-*uniform* in S if it is ϵ-close to the uniform ensemble over S (we sometimes omit S when it is clear from the context). $X = \{X_k\}_k$, $Y = \{Y_k\}_k$ are *computationally indistinguishable* if every poly(k)-time adversary $\mathcal{A}$ has negligible *distinguishing advantage*:

$$\text{Dist}_{X,Y}\text{Adv}[\mathcal{A}] = |\Pr[\mathcal{A}(X_k) = 1] - \Pr[\mathcal{A}(Y_k) = 1]| = \text{negl}(k) \ .$$

We often abbreviate and write $\text{DistAdv}[\mathcal{A}]$ when X, Y are clear from the context.

2.1 KDM Security

A public-key encryption scheme $\mathcal{E} = (G, E, D)$ is defined by its key generation, encryption and decryption algorithms. The key generation algorithm G takes as input the unary vector 1^k, where k is called the *security parameter* of the scheme. All other parameters of the scheme are parameterized by k. We let $\mathcal{S} = \{\mathcal{S}_k\}$ denote the space of secret keys and $\mathcal{M} = \{\mathcal{M}_k\}$ denote the message space of the encryption scheme. We refer the reader to [Gol04] for a formal definition of encryption schemes and their security.

In the scenario of key-dependent messages, we wish to model the case where functions of the secret key can be encrypted, and require that the resulting ciphertexts are indistinguishable from encryptions of 0. We want our definition to apply also for the case of "key cycles" where a function of one user's secret key is encrypted by another's public key and vice versa. The most inclusive definition, therefore, is parameterized by the number of users n and allows encrypting a function of the entire vector of n secret keys under any of the corresponding public keys (this is sometimes referred to as "clique security"). An additional parameter to be considered is the set of functions of the secret key that we allow to encrypt. We use the definition presented in [BHHO08].

Formally, let $\mathcal{E} = (G, E, D)$ be a public key encryption scheme, $n > 0$ be an integer, $\mathcal{S} = \{\mathcal{S}_k\}$ be the space of secret keys, and let $\mathcal{F} = \{\mathcal{F}_k\}$ be a class of functions such that $\mathcal{F}_k \subseteq \mathcal{S}_k^n \to \mathcal{M}_k$.

We define the $\text{KDM}^{(n)}$ game, w.r.t. the function class $\mathcal{F}$, played between a challenger and an adversary as follows.

Initialize. The challenger selects $b \xleftarrow{\$} \{0,1\}$ and generates, for all $i \in [n]$, key pairs $(sk_i, pk_i) \xleftarrow{\$} G(1^k)$. The challenger then sends $\{pk_i\}_{i\in[n]}$ to the adversary.

Query. The adversary makes queries of the form $(i, f) \in [n] \times \mathcal{F}_k$. For each query, the challenger computes $y \leftarrow f(sk_1, \ldots, sk_n)$ and sends the following ciphertext to the adversary.

$$c \leftarrow \begin{cases} E_{pk_i}(y) & \text{if } b = 0 \\ E_{pk_i}(0) & \text{if } b = 1. \end{cases}$$

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

The scheme $\mathcal{E}$ is $\text{KDM}^{(n)}$ secure if any polynomial time adversary $\mathcal{A}$ has negligible advantage:

$$\text{KDM}^{(n)}\text{Adv}[\mathcal{A}] = \left|\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]\right| = \text{negl}(k) \ .$$

We sometime denote $\text{KDM}^{(n)}_{\mathcal{F}}$ to indicate the function class in discussion.

2.2 Leakage Resilient Encryption

In the scenario of key-leakage resiliency, we wish to model the case where some (adversarially selected restricted amount of) information about the secret key is revealed to the attacker. We require that even in the presence of this additional information, the security of the scheme remains intact. The definition below is essentially adopted from [NS09].

Let $\mathcal{E} = (G, E, D)$ be a public key encryption scheme with key-space $\mathcal{S} = \{\mathcal{S}_k\}$ and message space $\mathcal{M} = \{\mathcal{M}_k\}$. We define the λ-leakage game, for the non-negative parameter $\lambda = \lambda(k)$, played between a challenger and an adversary as follows.

Initialize. The challenger selects $b \xleftarrow{\$} \{0,1\}$ and generates a key-pair $(sk, pk) \xleftarrow{\$} G(1^k)$. The challenger sends pk to the adversary.

Leakage. The adversary sends an efficiently computable function $f : \mathcal{S}_k \to \{0,1\}^\lambda$ to the challenger. The challenger computes $f(sk)$ and returns this value to the adversary.

Challenge. The adversary sends $m_0, m_1 \in \mathcal{M}_k$ to the challenger. The challenger computes $y \leftarrow E_{pk}(m_b)$ and sends y to the adversary.

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

The scheme $\mathcal{E}$ is λ-leakage secure if for any polynomial time adversary $\mathcal{A}$ it holds that

$$\text{Leak}_\lambda\text{Adv}[\mathcal{A}] = \left|\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]\right| = \text{negl}(k) \ .$$

We note that the definition can be extended to the case of chosen ciphertext attacks (see [NS09]), in this work we only consider the case of chosen plaintext attacks described above.

2.3 Auxiliary-Input Resilient Encryption

The scenario of auxiliary-input resiliency is quite similar to that of key-leakage resiliency described in Section 2.2. As in the previous case, we model a scenario where the attacker can access additional information about the secret key. In this case, however, the restriction on the amount of information is *computational* rather than information theoretic. The definition is adopted (and slightly adapted) from [DGK+10].

Let $\mathcal{E} = (G, E, D)$ be a public key encryption scheme with secret-key space $\mathcal{S} = \{\mathcal{S}_k\}$, public key space $\mathcal{P} = \{\mathcal{P}_k\}$ and message space $\mathcal{M} = \{\mathcal{M}_k\}$. For any family of functions $f = \{f_k : \mathcal{S} \times \mathcal{P} \to \{0,1\}^*\}$, we define the *inverting advantage* and *weak inverting advantage* of an adversary $\mathcal{A}$ as follows.

$$\begin{aligned} \mathrm{Inv}_f\mathrm{Adv}[\mathcal{A}] &= \Pr_{(sk,pk)\leftarrow G(1^k)}[\mathcal{A}(1^k, f_k(sk,pk)) = sk] \ , \\ \mathrm{Inv}_f^{\mathrm{weak}}\mathrm{Adv}[\mathcal{A}] &= \Pr_{(sk,pk)\leftarrow G(1^k)}[\mathcal{A}(1^k, pk, f_k(sk,pk)) = sk] \ . \end{aligned}$$

The public parameters of the scheme are an additional implicit argument to both f and $\mathcal{A}$ (in both definitions). Let ℓ denote the length of the binary representation of the secret key. A polynomial time computable function f is $\epsilon = \epsilon(\ell)$-hard to invert (resp. ϵ-weakly hard to invert)[6] if for any polynomial time $\mathcal{A}$ it holds that $\mathrm{Inv}_f\mathrm{Adv}[\mathcal{A}] \leq \epsilon$ (resp. $\mathrm{Inv}_f^{\mathrm{weak}}\mathrm{Adv}[\mathcal{A}] \leq \epsilon$). We stress that the notion of "hard to invert functions" is weaker than the standard notion of one-way functions since we require the recovery of the original secret key (and not of just any pre-image). Thus a function that is not one-way can still be hard to invert.

For any efficiently computable function family f, we consider the f-auxiliary input game, played between a challenger and an adversary as follows.

Initialize. The challenger selects $b \xleftarrow{\$} \{0,1\}$ and generates a key-pair $(sk, pk) \xleftarrow{\$} G(1^k)$. The challenger sends pk to the adversary.

Auxiliary input. The challenger computes $z \leftarrow f_k(sk, pk)$ and sends z to the adversary.

Challenge. The adversary sends $m_0, m_1 \in \mathcal{M}_k$ to the challenger. The challenger computes $y \leftarrow E_{pk}(m_b)$ and sends y to the adversary.

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

The scheme $\mathcal{E}$ is *ϵ-auxiliary input secure* (Aux_ϵ-secure) if for any ϵ-uninvertible f and for any polynomial time $\mathcal{A}$ it holds that

$$\mathrm{Aux}_f\mathrm{Adv}[\mathcal{A}] = \left|\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]\right| = \mathrm{negl}(k) \ .$$

$\mathcal{E}$ is *ϵ-weakly auxiliary-input secure* ($\mathrm{Aux}_\epsilon^{\mathrm{weak}}$-secure) if the above holds for any ϵ-weakly uninvertible function f.

2.4 Technical Tools

We use the following simple lemma.

[6] Note that we measure the hardness of f relative to its input length ℓ and not the security parameter k.

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

Proof. Define $d = (T \bmod N)$, then conditioned on the event $x \in [T-d]$, it holds that $(x \bmod N)$ is uniform in $\mathbb{Z}_N$. Therefore $(x \bmod N)$ is $(d/T) \leq (N/T)$-uniform. □

2.4.1 Simplified Leftover Hash Lemma and Applications

We use the following lemma which is an immediate corollary of the leftover hash lemma and explicitly appears in [BHHO08, 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 $H \times Y$.*

We also require the following consequence.

Lemma 2.3. *Let H be a 2-universal hash family from a set X to a set Y. Let $f : X \to Z$ be some function. Then the distribution $(h, h(x), f(x))$ where $h \xleftarrow{\$} H$, $x \xleftarrow{\$} X$ is $\sqrt{\frac{|Y|\cdot|Z|}{4|X|}}$-close to $(h, y, f(x))$, for $y \xleftarrow{\$} Y$.*

Proof. For all $z \in Z$, denote $X_z = \{x \in X : f(x) = z\}$ and $p_z = |X_z| / |X|$. Then by Lemma 2.2, $(h, h(x), f(x))$ conditioned on $f(x) = z$ is $\sqrt{\frac{|Y|}{4|X_z|}}$-close to $(h, y, f(x))$ conditioned on $f(x) = z$. Averaging over all $z \in Z$, we get that the distance between $(h, h(x), f(x))$ and $(h, y, f(x))$ is at most

$$\sum_{z \in Z} \left(p_z \cdot \sqrt{\frac{|Y|}{4\,|X_z|}} \right) = \sqrt{\frac{|Y|}{4\,|X|}} \cdot \sum_{z \in Z} \sqrt{p_z} \leq \sqrt{\frac{|Y|\,|Z|}{4\,|X|}} \ .$$

The result follows. □

We often use a families of 2-universal hash functions of the form presented below.

Lemma 2.4. *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.*

Note that the group $\mathbb{G}$ needs not be cyclic.

Proof. Consider $\mathbf{x} \neq \mathbf{y}$ and assume w.l.o.g that $x_1 \neq y_1$. Then for $g_1, \ldots, g_\ell \xleftarrow{\$} \mathbb{G}$ it holds that $g_1^{x_1 - y_1}$ is uniformly distributed in $\mathbb{G}$ and the result follows. □

2.4.2 The Goldreich-Levin Theorem

The Goldreich-Levin hard core predicate is stated in the following theorem.

Theorem 2.5 ([GL89])**.** *Let $f : \{0,1\}^n \to \{0,1\}^*$ be any (possibly randomized) function and let $\mathcal{A}$ be such that*

$$\left| \Pr_{\mathbf{x},\mathbf{r} \xleftarrow{\$} \{0,1\}^n} [\mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle) = 1] - \Pr_{\substack{\mathbf{x},\mathbf{r} \xleftarrow{\$} \{0,1\}^n, \\ u \xleftarrow{\$} \{0,1\}}} [\mathcal{A}(f(\mathbf{x}), \mathbf{r}, u) = 1] \right| \geq \epsilon \ ,$$

then there exists $\mathcal{B}^{\mathcal{A}}$ that runs in time $\mathrm{poly}(n, 1/\epsilon)$ such that

$$\Pr[\mathcal{B}^{\mathcal{A}}(f(\mathbf{x})) = \mathbf{x}] \geq \Omega(\epsilon^3/n) \ .$$

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.

3.2.1 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.[7] This leads to the immediate instantiation of the SG assumption by setting $\mathbb{G}_U = \mathbb{J}_N$, $\mathbb{G}_M = \{\pm 1\}$, $\mathbb{G}_L = \mathbb{QR}_N$, $h = (-1)$, $T = N \geq 2N'$.

3.2.2 Instantiation Under the DCR Assumption

The *decisional composite residuosity* (DCR) assumption, introduced by Paillier [Pai99], 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.[8]

4 Description of the Encryption Scheme

In the interest of clarity, we only present the QR-based scheme here. The general case is presented in Appendix B.1. The scheme $\mathcal{E}[\ell]$ is defined below.

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.

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

[7] 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.

[8] If greater efficiency is desired, we can use a generalized form of the assumption, presented in [DJ01]. Let $d \geq 1$ be a parameter that is polynomial in the security parameter and consider the group $\mathbb{G}_U = \mathbb{Z}^*_{N^{d+1}}$. Then $\mathbb{G}_U$ can be written as a product $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ where $\mathbb{G}_M$ is cyclic and has order $M = N^d$ and generator $h = (N+1)$. The group $\mathbb{G}_L$ is the group $\{x^{N^d} : x \in \mathbb{Z}^*_{N^{d+1}}\}$ which is isomorphic to $\mathbb{Z}^*_N$. Clearly $\gcd(N^d, \varphi(N)) = 1$ and we can use the bound $T = N^{d+1} \geq N^d \cdot \varphi(N)$.

It is proven in [DJ01] that under the DCR assumption, the subgroup indistinguishability problem defined by the above groups is hard for any polynomial d. Specifically, taking $d = 1$ gives Paillier's original assumption.

algorithm runs as follows: it samples $r \xleftarrow{\$} [N^2]$,[9] 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 the general definition in Appendix B.2.

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

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

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*

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

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 $|\Pr[H_1] - \Pr[H_0]| \leq 2 \cdot \mathrm{QRAdv}[\mathcal{B}]$.

[9] 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 mod $2N'$) is uniform in $\mathbb{Z}_{2N'}$, and this is achieved by sampling r from a much larger domain.

We further remark that, as pointed out to us by Adi Shamir, we could alternatively use $r \xleftarrow{\$} [(N-1)/2]$, since $U([(N-1)/2])$ and $U([\varphi(N)/2])$ are statistically indistinguishable.

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, the argument here 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

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

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

6 KDM Security

In this section, we discuss the KDM-security related properties of our scheme. We only discuss our QR-based encryption scheme in this section, in the interest of clarity. We prove the 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 and prove that for $\ell \geq n \cdot \log N + \omega(\log k)$, $\mathcal{E}[\ell]$ is also KDM$^{(n)}$-secure. Finally, extensions beyond affine functions are stated without proof in Section 6.3.

A presentation and analysis of the general case, including all relevant proofs, is provided in Appendix B.3.

Throughout this section we define $\mathcal{F}_{\text{aff}}$ to be the class of affine functions over $\mathbb{Z}_2$, namely the class of all functions of the form $f_{a_0,\mathbf{a}}(\mathbf{x}) = a_0 + \sum a_i x_i$, where $a_i, x_i \in \mathbb{Z}_2$, and arithmetics are also over $\mathbb{Z}_2$.

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 $\mathrm{KDM}^{(1)}_{\mathcal{F}_{\mathrm{aff}}}$-adversary for $\mathcal{E}[\ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that*

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

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

Proof. The proof proceeds by a series of hybrids.

Hybrid H_0. This hybrid is identical to the $\mathrm{KDM}^{(1)}$ game with $b=0$. By definition $\Pr_{H_0}[b'=1] = \Pr[b'=1|b=0]$.

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 $f_{a_0,\mathbf{a}} \in \mathcal{F}_{\mathrm{aff}}$ 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

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

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 2.4 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 the $\mathrm{KDM}^{(1)}$ game with $b=1$, as all queries are answered by encryptions of 0: $\Pr_{H_4}[b'=1] = \Pr[b'=1|b=1]$. Summing the terms above, the result follows (where $\mathcal{B}$ is, say, a weighted average between $\mathcal{B}_1$ and $\mathcal{B}_2$). □

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

A formal statement and proof for the QR case follows. For the statement and proof in the general case, see Appendix B.3.3.

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*

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

The theorem implies that taking $\ell = n\cdot\log N + \omega(\log k)$ is sufficient for $\mathrm{KDM}^{(n)}$-security.

Proof. Let us first introduce the notation used in the proof: We now consider functions in $\mathcal{F}_{\mathrm{aff}}$ that are applied to a concatenated vector of n secret keys. We will denote such functions by $f_{a_0,\mathbf{a}_1,\ldots,\mathbf{a}_n}$, where $a_0\in\{0,1\}$ and $\mathbf{a}_i\in\{0,1\}^\ell$, and such that for all $\mathbf{x}_1,\ldots,\mathbf{x}_n\in\{0,1\}^\ell$ the function is defined by

$$f_{a_0,\mathbf{a}_1,\ldots,\mathbf{a}_n}(\mathbf{x}_1,\ldots,\mathbf{x}_n) = a_0 + \sum_{j\in[n],i\in[\ell]} a_{j,i}\cdot x_{j,i}\ .$$

All arithmetic operations are over $\mathbb{Z}_2$.

The proof follows the outline of the proof of Theorem 6.1 with slight modifications to the hybrids.

Hybrid H_0. This hybrid is identical to the $\mathrm{KDM}^{(n)}$ game with $b=0$. Let $\{sk_i=\mathbf{s}_i\}_{i\in[n]}$ denote the generated secret keys and $\{pk_i=(g_{0,i},\mathbf{g}_i)\}_{i\in[n]}$ denote the public keys produced by the challenger. By definition $\Pr_{H_0}[b'=1]=\Pr[b'=1|b=0]$.

Hybrid H_1. We change the way the challenger answers the adversary's queries. For each query $((a_0,\{\mathbf{a}_j\}_{j\in[n]}),i)$ made by $\mathcal{A}$, the challenger does as follows.

Define $\mathbf{y}_{i,j}=\mathbf{s}_i\oplus\mathbf{s}_j$ (the binary XOR operation). Given $\{\mathbf{y}_{i,j}\}_{i,j}$, the challenger finds $(a'_0,\mathbf{a}')$ such that $f_{a'_0,\mathbf{a}'}(\mathbf{s}_i) = f_{a_0,\{\mathbf{a}_j\}_{j\in[n]}}(\mathbf{s}_1,\ldots,\mathbf{s}_n)$. This is possible to do without knowing the values of the $\{\mathbf{s}_i\}$, only $\{\mathbf{y}_{i,j}\}$: Consider the element $s_{j,i'}$ (the i'^{th} element of $\mathbf{s}_j$). We know that $s_{j,i'} = s_{i,i'}\oplus(\mathbf{y}_{i,j})_{i'}$. Therefore we know that if $(\mathbf{y}_{i,j})_{i'}=0$ then $s_{j,i'}=s_{i,i'}$ and if $(\mathbf{y}_{i,j})_{i'}=1$ then $s_{j,i'}=1-s_{i,i'}$. We can thus replace the $a_{j,i'}\cdot s_{j,i'}$ element in the description of the function f with either $a_{j,i'}\cdot s_{i,i'}$ or $a_{j,i'}\cdot(1-s_{i,i'})$, depending on the (known) value of $(\mathbf{y}_{i,j})_{i'}$. Doing this one variable at a time, results in an affine function of only $\mathbf{s}_i$. We again stress that we only used $\{\mathbf{y}_{i,j}\}_{i,j}$ for this transformation.

The challenger in this hybrid answers with $(c_0,\mathbf{c}) = ((-1)^{a'_0}\cdot g_{0,i}^r, (-1)^{\mathbf{a}'}\cdot\mathbf{g}_i^r)$ instead of $(c_0,\mathbf{c}) = ((-1)^{a'_0+\sum_{j\in[\ell]}a'_j\cdot s_{i,j}}\cdot g_{0,i}^r, \mathbf{g}_i^r)$.

It holds that

$$\left|\Pr_{H_1}[b'=1] - \Pr_{H_0}[b'=1]\right| \leq 4nt\ell\cdot\mathrm{QRAdv}[\mathcal{B}_1] + O(nt\ell/N)\ ,$$

since, as in the proof of Theorem 6.1, the difference between the hybrids can be viewed as a t-round interactive $(n\ell)$-vector game, considering $(\mathbf{g}_1,\ldots,\mathbf{g}_n)$ as the first message, and now the simulated adversary only needs a part of the answer for each query (the one that is respective to the user i for which the query was made).

Hybrid H_2. Following the proof outline of Theorem 6.1, we change the distributions of $g_{0,i}$ for *all* $i\in[n]$, to $U(\mathbb{QR}_N)$. The challenger still needs to know $\{\mathbf{y}_{i,j}\}_{i,j\in[n]}$ in order to answer the queries

so we need to prove that even fixing $\{\mathbf{y}_{i,j}\}_{i,j\in[n]}$, hybrid H_2 is close to H_1. Intuitively speaking, this will require us to "extract" n uniform elements of $\mathbb{QR}_N$ out of a single $\mathbf{s}$ (because once one of them is specified, all others are determined by the values of $\mathbf{y}_{i,j}$). Therefore, for security to hold, we have to require that ℓ is proportional to n.

We now wish to apply Lemma 2.2 to claim that the hybrids are statistically close. To do that, we consider the following family of hash functions (defined for a fixed value of $\{\mathbf{y}_{i,j}\}$)

$$z_{\mathbf{g}_1,\ldots,\mathbf{g}_n}(\mathbf{s}_1) = \left(\prod g_{1,i}^{s_{1,i}}, \prod g_{2,i}^{s_{1,i}\oplus(\mathbf{y}_{1,2})_i}, \ldots, \prod g_{n,i}^{s_{1,i}\oplus(\mathbf{y}_{1,n})_i}\right) .$$

This family is 2-universal (by a similar argument to Lemma 2.4, using the fact that the vectors $\mathbf{g}_i$ are independent). The output describes the distribution of $g_{0,i}$ in the case where all $\mathbf{y}_{i,j}$ are known, but $\mathbf{s}_1$ is not. We can now apply Lemma 2.2 respective to this family and conclude that

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

Hybrid H_3. Note that at this point the public keys are distributed uniformly and independently of $\mathbf{y}_{i,j}$'s. We again change the way the challenger answers queries, along the lines of the proof of Theorem 6.1. Instead of answering with $(c_0,\mathbf{c}) = ((-1)^{a'_0}\cdot g_{0,i}^r, (-1)^{\mathbf{a}'}\cdot\mathbf{g}_i^r)$, the challenger now answers with $(c_0,\mathbf{c}) = (g_{0,i}^r, \mathbf{g}_i^r)$. This can be viewed as a t-round interactive $n(\ell+1)$-vector game (similarly to the previous hybrid) and thus

$$\left|\Pr_{H_3}[b'=1] - \Pr_{H_2}[b'=1]\right| \leq 4nt(\ell+1)\cdot \mathrm{QRAdv}[\mathcal{B}_2] + O(nt\ell/N) .$$

Hybrid H_4. We revert the distributions of the $g_{0,i}$'s to the original one. As in hybrid H_2 we have $|\Pr_{H_4}[b'=1] - \Pr_{H_3}[b'=1]| \leq \frac{1}{2}\cdot(N\cdot 2^{-\ell/n})^{n/2}$.

Hybrid H_4 is identical to the $\mathrm{KDM}^{(n)}$ game with $b=1$ as all queries are answered by encryptions of 0 and the claim follows. □

6.3 Beyond Affine Functions

Two building blocks have been suggested in [BGK09, BHHI09] 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.

The first corollary is derived using [BGK09, 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 6.3. *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* $\mathrm{KDM}^{(1)}$*-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 [BHHI09, Theorem 4.1].

Corollary 6.4. *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).*

These results can be generalized to any SG assumption. For the generalized statements and proofs, see Appendix B.3.4.

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 $\boldsymbol{\sigma} \xleftarrow{\$} \mathbb{Z}_2^\ell$. The views of the adversary in the two cases are indistinguishable (by an interactive vector game).[10] 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.[11] The result in the QR case is formally stated and proven below. The general result is stated and proven in Appendix B.4.

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

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

Proof. We prove by a series of hybrids (experiment). Each experiment defines a binary random variable (one can think of a value of 1 as a "success" in the experiment).

Hybrid H_0. This hybrid describes the following experiment: a challenger flips a coin $b \xleftarrow{\$} \{0,1\}$ and simulates the λ-leakage game with $\mathcal{A}$. It returns 1 if and only if $b' = b$, where b' is the value returned by $\mathcal{A}$. By definition

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

Hybrid H_1. We change the encryption algorithm. In this hybrid, we encrypt the message m_b by first computing $\mathbf{c} = \mathbf{g}^r$ and then using $\mathbf{s}$ to produce $c_0 = (-1)^{m_b} \cdot \prod_{i \in [\ell]} c_i^{-s_i}$. The ciphertext distribution does not change and hence $\Pr[H_1 = 1] = \Pr[H_0 = 1]$.

Hybrid H_2. Again we change the encryption. This time the challenger samples $\boldsymbol{\sigma} \xleftarrow{\$} \{0,1\}^\ell$ and uses $\mathbf{c} = (-1)^{\boldsymbol{\sigma}} \cdot \mathbf{g}^r$ instead of $\mathbf{c} = \mathbf{g}^r$. Note that the difference between H_1 and H_2 is exactly a 1-round interactive ℓ-vector game and thus by Lemma 5.1, there exists an adversary $\mathcal{B}$ such that

$$|\Pr[H_2 = 1] - \Pr[H_1 = 1]| \leq 4\ell \cdot \mathrm{QRAdv}[\mathcal{B}] + O(\ell/N) \ .$$

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

[11] In the spirit of [NS09], 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.

Hybrid H_3. We notice that in H_2, the distribution of c_0 is

$$c_0 = (-1)^{m_b} \cdot \prod_{i\in[\ell]} (-1)^{s_i\cdot\sigma_i} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r = (-1)^{m_b+\sum s_i\cdot\sigma_i} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r \ .$$

In hybrid H_3 we change this distribution. The challenger samples $u \xleftarrow{\$} \{0,1\}$ and sets

$$c_0 = (-1)^{m_b+u} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r \ .$$

To analyze this hybrid, we recall that $\prod_{i\in[\ell]} g_i^{-s_i} \in \mathbb{QR}_N$ and use Lemma 2.4 and Lemma 2.3 to conclude that even for a given $\mathbf{g}$ it holds that $(\boldsymbol{\sigma}, \sum s_i\sigma_i, \prod_{i\in[\ell]} g_i^{-s_i}, f(\mathbf{s}))$ is $\frac{1}{2}\cdot\sqrt{N\cdot 2^{\lambda-\ell}}$-close to $(\boldsymbol{\sigma}, u, \prod_{i\in[\ell]} g_i^{-s_i}, f(\mathbf{s}))$. It follows that

$$|\Pr[H_3=1] - \Pr[H_2=1]| \le \frac{1}{2}\cdot\sqrt{N\cdot 2^{\lambda-\ell}} \ .$$

Hybrid H_4. We further change c_0 and now set it to be

$$c_0 = (-1)^{u} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r \ .$$

Since u is uniform, it is distributed identically to $m_b + u$ (note that the arithmetics here are over $\mathbb{Z}_2$) and thus $\Pr[H_4=1] = \Pr[H_3=1]$. In H_4, however, the ciphertext distribution is independent of b. Therefore $\Pr[H_4=1] = \frac{1}{2}$. Combining all of the above, the result follows. □

8 Auxiliary-Input Resiliency

As in previous work, we start by proving weak auxiliary-input security in Lemma 8.1 below and then derive general auxiliary-input security for sub-exponentially hard functions in Corollary 8.2. The complete proofs for the general case appear in Appendix B.5.

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). Let $\mathcal{A}$ be an f-auxiliary input adversary for $\mathcal{E}[\ell]$. Then there exists an adversary $\mathcal{B}$ such that*

$$\mathrm{Aux}_f^{\mathrm{weak}}\mathrm{Adv}[\mathcal{A}] \le 8\ell\cdot\mathrm{QRAdv}[\mathcal{B}] + O(\ell/N) + \mathrm{negl}(k) \ .$$

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 only provide a proof sketch of Lemma 8.1, for a full proof (for the general case) see Lemma B.10 in Appendix B.5.

Proof sketch. The proof is almost identical to that of Theorem 7.1. The only difference is that now we argue that $|\Pr[H_3=1] - \Pr[H_2=1]| = \mathrm{negl}(k)$ by applying Theorem 2.5 (the Goldreich-Levin theorem) to the uninvertible function $f'(\mathbf{s}) = (\mathbf{g}, \prod g_i^{-s_i}, f(\mathbf{s}))$. □

An immediate corollary (see [DGK+10, Lemma 4]) enables us to state that $\mathcal{E}[\ell]$ is (ϵ/N)-auxiliary input secure for any negligible ϵ, this is because the only part of the public key that depends on the secret key is g_0, whose value can be "guessed" with probability $1/N$. For a formal proof, see Corollary B.11 in Appendix B.5. Note that in order for $(\mathrm{negl}(k)/N)$-hard to invert functions to even exist, it must be that $\ell \geq \log N + \omega(\log k)$, since any function of ℓ input bits is trivially invertible with probability at least $2^{-\ell}$.

We can derive the following corollary (for proof, see Corollary B.12 in Appendix B.5).

Corollary 8.2. *Assuming that a subgroup indistinguishability assumption holds, then for any constant $\delta > 0$ there exists a $2^{-\ell^\delta}$-auxiliary input resilient encryption scheme.*

Acknowledgments

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

References

[ABHS05] 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.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Halevi [Hal09], pages 595–618.

[AGV09] 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.

[BGK09] Zvika Brakerski, Shafi Goldwasser, and Yael Kalai. Circular-secure encryption beyond affine functions. Cryptology ePrint Archive, Report 2009/485, 2009. `http://eprint.iacr.org/`.

[BHHI09] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. Cryptology ePrint Archive, Report 2009/511, 2009. `http://eprint.iacr.org/`.

[BHHO08] 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.

[BRS02] 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.

[CL01] 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.

[CS02] 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.

[DGK+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In TCC 2010 (to appear), 2010.

[DJ01] 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.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *STOC*, pages 621–630. ACM, 2009.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. ACM, 1989.

[GM82] 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.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol04] Oded Goldreich. *Foundations of Cryptography - Basic Applications*. Cambridge University Press, 2004.

[Hal09] 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.

[HO09] 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/`.

[HSH+08] 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.

[KPSY09] 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.

[LC03] 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.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [Hal09], pages 18–35.

[Pai99] 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.

[PW08] 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.

[Reg05] 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.

A A Generalized Goldreich-Levin Theorem

We use a generalized version of Theorem 2.5, essentially adopted from [DGK+10, Theorem 1] and slightly adapted (see explanation below).

Theorem A.1 (adapted from [DGK+10, Theorem 1])**.** *Let $f : \{0,1\}^n \to \{0,1\}^*$ be any (possibly randomized) function, let $K \in \mathbb{N}$, $K > 1$ and let $\mathcal{A}$ be such that*

$$\left| \Pr[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle) = 1] - \Pr[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, u) = 1] \right| \geq \epsilon \ ,$$

where $\mathbf{x} \xleftarrow{\$} \{0,1\}^n, \mathbf{r} \xleftarrow{\$} \mathbb{Z}_K^n, u \xleftarrow{\$} \mathbb{Z}_K$ and the inner product is over $\mathbb{Z}_K$, then there exists $\mathcal{B}^{\mathcal{A}}$ that runs in time $\mathrm{poly}(n, 1/\epsilon)$ such that

$$\Pr[\mathcal{B}^{\mathcal{A}}(f(\mathbf{x})) = \mathbf{x}] \geq \frac{\epsilon}{8 \cdot K^{1+\log(8n/\epsilon^2)}} \ .$$

Furthermore, $\mathcal{B}$ only needs to sample uniformly in $\mathbb{Z}_K$ and to add two elements in $\mathbb{Z}_K$ and does not use any other property of $\mathbb{Z}_K$.

This theorem is different from [DGK+10, Theorem 1] in a few points. We allow K to take any value (so long as operations over $\mathbb{Z}_K$ are efficient), while they restricted their attention to prime K. This has a technical effect in the proof since the original proof assumed (implicitly) that it was easy to find a polynomial set of elements $\{\rho_i\} \subseteq \mathbb{Z}_K$ such that for all $i \neq j$, $(\rho_i - \rho_j) \in \mathbb{Z}_K^*$ (namely, is a unit in the ring $\mathbb{Z}_K$). This is easy to achieve in the of prime K, but for a general K whose factorization (and perhaps even its exact value) is unknown, this is not necessarily the case. Therefore in our proof we only use the trivial set $\{0,1\}$, which implies worse parameters. Specifically the power of K in the success probability of the invertor, which is the dominant factor, is logarithmic in our statement but constant (specifically, 2) in theirs. We remark that if $\mathbb{Z}_K$ is such that finding a non-unit is computationally hard (one example is $\mathbb{Z}_N$ for an RSA number N), then similar parameters to [DGK+10, Theorem 1] can be achieved, using the same techniques.

An additional change is that we only allow $f : \{0,1\}^n \to \{0,1\}^*$ while their theorem applies to $f : H^n \to \{0,1\}^*$, for any set $H \subseteq \mathbb{Z}_K$ of polynomial cardinality. This change is partly to simplify the proof (since we only use the theorem for binary f) and partly because for a non-prime K, the requirement we need to impose on H and the affect on the parameters seem to make this even-more-general version not useful.

The formal proof follows.

Proof. Given an algorithm $\mathcal{A}$ as stated in the theorem, it holds that

$$\Pr_{\mathbf{x}} \left[\left| \Pr_{\mathbf{r}}[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle) = 1] - \Pr_{\mathbf{r},u}[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, u) = 1] \right| > \epsilon/2 \right] \geq \epsilon/2 \ .$$

Our reduction $\mathcal{B}^{\mathcal{A}}$ will succeed with sufficiently high probability for $y = f(\mathbf{x})$ for which

$$\Pr_{\mathbf{r}}[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle) = 1] - \Pr_{\mathbf{r},u}[\mathcal{A}(f(\mathbf{x}), \mathbf{r}, u) = 1] > \epsilon/2 \ .$$

Note that we removed the absolute value, which incurs a factor of $1/2$ in the final success probability. From now on, condition on this event.

The reduction $\mathcal{B}^{\mathcal{A}}$, on an input $y = f(\mathbf{x})$, runs as follows. Define $m = 8n/\epsilon^2$ and let $c = 1 + \lfloor \log m \rfloor$. Sample $\mathbf{z}_1, \ldots, \mathbf{z}_c \xleftarrow{\$} \mathbb{Z}_K^n$ and $g_1, \ldots, g_c \xleftarrow{\$} \mathbb{Z}_K$. With probability $1/K^c \geq 1/K^{1+\log(8n/\epsilon^2)}$ it holds that for all $j \in [c]$, $\langle \mathbf{z}_j, \mathbf{x} \rangle = g_j$. From now on, condition on this event as well.

For all $\boldsymbol{\rho} \in \{0,1\}^c \setminus \{\mathbf{0}\}$, define $\mathbf{r}_{\boldsymbol{\rho}} = \sum_{j \in [c]} \rho_j \cdot \mathbf{z}_j$ and $h_{\boldsymbol{\rho}} = \sum_{j \in [c]} \rho_j \cdot g_j$. Then $\{\mathbf{r}_{\boldsymbol{\rho}}\}$ are uniformly distributed, pairwise independent (this is where we use the fact that ± 1 are units in $\mathbb{Z}_K$) and it holds that $\langle \mathbf{r}_{\boldsymbol{\rho}}, \mathbf{x} \rangle = h_{\boldsymbol{\rho}}$.

Next, it samples $\tau_{\boldsymbol{\rho}} \xleftarrow{\$} \mathbb{Z}_K$ for all $\boldsymbol{\rho}$ and computes, for all $i \in [n]$ and $\boldsymbol{\rho} \in \{0,1\}^c \setminus \{0\}$,

$$s_{i,\boldsymbol{\rho}} = \mathcal{A}(y, \mathbf{r}_{\boldsymbol{\rho}} + \tau_{\boldsymbol{\rho}} \cdot \mathbf{e_i}, h_{\boldsymbol{\rho}} + \tau_{\boldsymbol{\rho}}) - \mathcal{A}(y, \mathbf{r}_{\boldsymbol{\rho}} + \tau_{\boldsymbol{\rho}} \cdot \mathbf{e_i}, h_{\boldsymbol{\rho}}) \ ,$$

it then computes $s_i = \mathbb{E}_{\boldsymbol{\rho}}[s_{i,\boldsymbol{\rho}}] = \frac{\sum_{\boldsymbol{\rho}} s_{i,\boldsymbol{\rho}}}{2^c - 1}$ (i.e. we consider the uniform distribution over all values of $\boldsymbol{\rho}$) and if $s_i \geq 0$, it sets $x_i = 1$, otherwise it sets $x_i = 0$.

To analyze, we notice that $s_{i,\boldsymbol{\rho}}$ is distributed like $\mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle) - \mathcal{A}(f(\mathbf{x}), \mathbf{r}, u)$ if $x_i = 1$ and like $\mathcal{A}(f(\mathbf{x}), \mathbf{r}, u) - \mathcal{A}(f(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle)$ if $x_i = 0$. Thus, for any fixed value of $\boldsymbol{\rho}$, $\mathbb{E}_{\mathbf{r}_{\boldsymbol{\rho}}}[s_{i,\boldsymbol{\rho}}] \geq \epsilon/2$ if $x_i = 1$ and $\mathbb{E}_{\mathbf{r}_{\boldsymbol{\rho}}}[s_{i,\boldsymbol{\rho}}] \leq -\epsilon/2$ if $x_i = 0$. Since $\{s_{i,\boldsymbol{\rho}}\}_{\boldsymbol{\rho}}$ are pairwise independent, we can apply Chebishev's inequality and get that with probability at least $1/(m \cdot (\epsilon/2)^2) \geq 1/(2n)$ it holds that s_i is within $\epsilon/2$ of its expected value, in which case x_i is computed correctly. Applying the union bound over all i, we have that $\mathbf{x}$ is computed correctly with probability $1/2$.

Combining all of the terms above, the result follows. □

B Constructions and Proofs for General Subgroup Indistinguishability Assumptions

B.1 Description of the Encryption Scheme

The scheme $\mathcal{E}[\mathbb{G}_U, \ell]$, which is a generalization of $\mathcal{E}[\ell]$ presented in Section 4, is defined as follows.

Parameters. The scheme is parameterized by a group $\mathbb{G}_U$, as described in Section 3. Namely, a probabilistic algorithm that given the security parameter 1^k, produces an instance $I_{\mathbb{G}_U}$ of $\mathbb{G}_U$.

An additional parameter is the value ℓ that is polynomial in the security parameter but its exact value is determined based on the specific application. The message space for the encryption scheme is $\mathcal{M} = \mathbb{G}_M$.

Key generation. The key generator first samples an instance $I_{\mathbb{G}_U}$ of subgroup indistinguishability. As in our QR-based scheme, the same instance can be used by all users and this it is sometimes treated as an implicit public parameter. It samples $\mathbf{s} \xleftarrow{\$} \{0,1\}^\ell$ and sets $sk = \mathbf{s}$.[12] It then samples $\mathbf{g} \xleftarrow{\$} {\mathbb{G}_L}^\ell$ 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 the instance $I_{\mathbb{G}_U}$ as an additional implicit public parameter.

Encryption. On inputs a public key $pk = (g_0, \mathbf{g})$ and a message $m \in \mathbb{G}_M$, the encryption algorithm runs as follows: it samples $r \xleftarrow{\$} [T^2]$ and computes $\mathbf{c} = \mathbf{g}^r$ and $c_0 = m \cdot g_0^r$. 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 $m = c_0 \cdot \prod_{i \in [\ell]} c_i^{s_i}$.

The completeness of the scheme in this case follows by definition as well.

[12] An alternative representation of the secret key can be to set $sk = h^{\mathbf{s}}$ and extract $\mathbf{s}$ via discrete logarithm (which is easy since $s_i \in \{0,1\}$) during the decoding process. This alternative representation will lead to a different representation of the class of functions for which we get KDM security. See Section B.3.1 for more details.

B.2 The Interactive Vector Game

The interactive ℓ-vector game, for a general SG assumption, is defined as follows (see Section 5 for an explicit presentation of the QR case).

Initialize. Let $\mathbb{G}_U$ be as defined in Section 3.1. The challenger samples $b \xleftarrow{\$} \{0,1\}$ and also generates an instance $I_{\mathbb{G}_U}$ and a vector $\mathbf{g} \xleftarrow{\$} {\mathbb{G}_L}^{\ell}$. It sends $I_{\mathbb{G}_U}$ and $\mathbf{g}$ to the adversary.

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

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

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

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

We show that under the SG assumption, no polynomial time adversary can obtain a non-negligible advantage in the game.

Lemma B.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* SG *such that*

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

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 the following 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 $\mathbf{a} \in {\mathbb{G}_M}^{\ell}$ with $(g_1^r, \ldots, g_{i-1}^r, a_i^b \cdot g_i^r, a_{i+1} \cdot g_{i+1}^r, \ldots, a_\ell \cdot g_\ell^r)$. 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 uniform in $\mathbb{G}_L$) with $h \cdot g_i$. We get that there exists $\mathcal{B}', \mathcal{B}$ such that $|\Pr[H_1] - \Pr[H_0]| \leq \mathrm{SG'Adv}[\mathcal{B}'] \leq 2 \cdot \mathrm{SGAdv}[\mathcal{B}]$.

We note that in this hybrid the adversary's query is answered with $(g_1^r, \ldots, g_{i-1}^r, a_i^b \cdot (h \cdot g_i)^r, a_{i+1} \cdot g_{i+1}^r, \ldots, a_\ell \cdot g_\ell^r)$.

Hybrid H_2. In this hybrid the only change is that now $r \xleftarrow{\$} \mathbb{Z}_{M \cdot L}$ rather than $U([T^2])$. By Lemma 2.1 it follows that $|\Pr[H_2] - \Pr[H_1]| \leq 1/T$.

We denote $r_1 = (r \bmod M)$ and $r_2 = (r \bmod L)$. By the Chinese Remainder Theorem it holds that r_1, r_2 are uniform in $\mathbb{Z}_M, \mathbb{Z}_L$ respectively and are independent. The answer to the query in this scenario is therefore

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

However since h^{r_1} is uniform in the cyclic group $\mathbb{G}_M$ then $a_i^b \cdot h^{r_1}$ is also uniform and is independent of b. It follows that $\Pr[H_2] = \frac{1}{2}$.

It follows that $\mathrm{IV}_\ell\mathrm{Adv}[\mathcal{A}_1] \leq 4\ell \cdot \mathrm{SGAdv}[\mathcal{B}] + 2\ell/T$, and the result follows. □

B.3 KDM Security

In this section, we state and prove the KDM-related properties of our general scheme $\mathcal{E}[\mathbb{G}_U, \ell]$. We start by defining the class $\mathcal{F}_{\text{aff}}$ of affine functions over $\mathbb{G}_M$ in Section B.3.1. Then, in Section B.3.2 we show that for $\ell \geq \log L + \omega(\log k)$, $\mathcal{E}[\mathbb{G}_U, \ell]$ is $\text{KDM}^{(1)}_{\mathcal{F}_{\text{aff}}}$-secure. We proceed, in Section B.3.3, to show that for $\ell \geq n \log L + \omega(\log k)$, it holds that $\mathcal{E}[\mathbb{G}_U, \ell]$ is $\text{KDM}^{(n)}_{\mathcal{F}_{\text{aff}}}$-secure. Section B.3.4 explains how to extend the KDM results of the preceding sections beyond affine functions, by using either the techniques of [BGK09] or those of [BHHI09]. This section contains the general versions and full proofs of the statements in Section 6.

B.3.1 The Class of Affine Functions

Recall that $\mathbb{G}_M$ is a cyclic group of order M with generator h. The class of affine functions over $\mathbb{G}_M$ is the class of affine functions over $\mathbb{Z}_M$ "in the exponent". If the discrete logarithm problem in h is easy, then the two are computationally equivalent. A formal definition follows.

The class $\mathcal{F}_{\text{aff}}$, for input dimension ℓ, is the set of functions $\mathcal{F}_{\text{aff}} = \{f_{\boldsymbol{\alpha},\beta} : \mathbb{Z}_M^\ell \to \mathbb{G}_M\}_{\beta \in \mathbb{G}_M, \boldsymbol{\alpha} \in \mathbb{G}_M{}^\ell}$ where $f_{\boldsymbol{\alpha},\beta}(\mathbf{x}) = \beta \cdot \prod_{i \in [\ell]} \alpha_i^{x_i}$. The function $f_{\boldsymbol{\alpha},\beta}$ is represented by $(\boldsymbol{\alpha}, \beta)$.

In Section B.3.3 we consider affine functions for input dimension $n \cdot \ell$, in which case we will "break" $\boldsymbol{\alpha}$ into n parts and present $\mathcal{F}_{\text{aff}} = \{f_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n,\beta} : (\mathbb{Z}_M^\ell)^n \to \mathbb{G}_M\}_{\beta \in \mathbb{G}_M, \boldsymbol{\alpha}_i \in \mathbb{G}_M{}^\ell}$ where $f_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n,\beta}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \beta \cdot \prod_{i \in [n], j \in [\ell]} \alpha_{i,j}^{x_{i,j}}$.

We remark that [BHHO08] gave a "dual" definition of this function class. In their formulation, the secret key of the encryption scheme is $h^{\mathbf{s}}$ and the decryption algorithm needs to recover $\mathbf{s}$ as a first step. A description of an affine function using their formulation contains a vector $\mathbf{a} \in \mathbb{Z}_M$ such that $h^{\mathbf{a}}$ corresponds to $\boldsymbol{\alpha}$ in our formulation. Our proofs (as well as theirs) work using both definitions, we chose to work with the above for aesthetic reasons.[13]

B.3.2 $\text{KDM}^{(1)}$-Security

We can now prove $\text{KDM}^{(1)}$-security for $\mathcal{E}[\mathbb{G}_U, \ell]$. The high level idea is the same as in the QR-based scheme as described in Section 6.1.

Theorem B.2. *Let $\mathcal{A}$ be a $\text{KDM}^{(1)}_{\mathcal{F}_{\text{aff}}}$-adversary for $\mathcal{E}[\mathbb{G}_U, \ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that*

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

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

Proof. The proof proceeds by a series of hybrids.

Hybrid H_0. This hybrid is identical to the $\text{KDM}^{(1)}$ game with $b = 0$. By definition $\Pr_{H_0}[b' = 1] = \Pr[b' = 1 | b = 0]$.

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 $(\boldsymbol{\alpha}, \beta) \in \mathcal{F}_{\text{aff}}$ was answered by $(c_0, \mathbf{c}) = (\beta \cdot \prod_{i \in [\ell]} \alpha_i^{s_i} \cdot g_0^r, \mathbf{g}^r)$. In hybrid H_1, it will be answered by $(c_0, \mathbf{c}) = (\beta \cdot g_0^r, \boldsymbol{\alpha} \cdot \mathbf{g}^r)$.

[13] We further remark that the if the group $\mathbb{G}_M$ is not efficiently recognizable, we may also require that the description of a function contains the logarithms of $\boldsymbol{\alpha}, \beta$ relative to h, in order to guarantee that the adversary doesn't "illegally" query a function outside of the prescribed class.

We prove that

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

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

To see this, we notice that in both hybrids $c_0 = \beta\cdot\prod_{i\in[\ell]}(\alpha_i\cdot c_i^{-1})^{s_i}$ 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 B.1, the result follows.

Hybrid H_2. In this hybrid, we change the distribution of g_0, which will now be sampled from $U(\mathbb{G}_L)$. By Lemma 2.4 combined with Lemma 2.2, $(g_0, \mathbf{g})$ is $\sqrt{\frac{L}{2^{\ell+2}}}$-uniform. Thus

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

Hybrid H_3. In this hybrid, we again change the way the challenger answers queries. Now instead of answering $(c_0, \mathbf{c}) = (\beta\cdot g_0^r, \boldsymbol{\alpha}\cdot\mathbf{g}^r)$, 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 B.1,

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

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

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

However, hybrid H_4 is identical to the $\mathrm{KDM}^{(1)}$ game with $b=1$ as all queries are answered by encryptions of 0: $\Pr_{H_4}[b'=1] = \Pr[b'=1|b=1]$. Summing the terms above, the result follows (where $\mathcal{B}$ is, say, a weighted average between $\mathcal{B}_1$ and $\mathcal{B}_2$). □

B.3.3 $\mathrm{KDM}^{(n)}$-Security

We go on to prove $\mathrm{KDM}^{(n)}$-security for our scheme. Unlike the schemes of [BHHO08, ACPS09], we do not achieve a single scheme that is secure w.r.t. any polynomial n. We do, however, prove that increasing the value of ℓ enables supporting more users. The result is stated in the following lemma.

Theorem B.3. *Let $\mathcal{A}$ be a $\mathrm{KDM}^{(n)}_{\mathcal{F}_{\mathrm{aff}}}$-adversary for $\mathcal{E}[\mathbb{G}_U, \ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that*

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

The theorem implies that taking $\ell = n\log L + \omega(\log k)$ is sufficient for $\mathrm{KDM}^{(n)}$-security.

Proof. The proof follows the outline of the proof of Theorem B.2 with slight modifications to the hybrids.

Hybrid H_0. This hybrid is identical to the $\mathrm{KDM}^{(n)}$ game with $b = 0$. Let $\{sk_i = \mathbf{s}_i\}_{i\in[n]}$ denote the generated secret keys and $\{pk_i = (g_{0,i}, \mathbf{g}_i)\}_{i\in[n]}$ denote the public keys produced by the challenger. By definition $\Pr_{H_0}[b' = 1] = \Pr[b' = 1|b = 0]$.

Hybrid H_1. We change the way the challenger answers the adversary's queries. For each query $((\{\boldsymbol{\alpha}_j\}_{j\in[n]}, \beta), i)$ made by $\mathcal{A}$, the challenger does as follows.

Define $\mathbf{y}_{i,j} = \mathbf{s}_i \oplus \mathbf{s}_j$ (the binary XOR operation). Given $\{\mathbf{y}_{i,j}\}_{i,j}$, the challenger finds $(\boldsymbol{\alpha}', \beta')$ such that $f_{\boldsymbol{\alpha}',\beta'}(\mathbf{s}_i) = f_{\{\boldsymbol{\alpha}_j\}_{j\in[n]},\beta}(\mathbf{s}_1, \ldots, \mathbf{s}_n)$. This is possible to do without knowing the values of the $\{\mathbf{s}_i\}$, only $\{\mathbf{y}_{i,j}\}$: Consider the element $s_{j,i'}$ (the i'^{th} element of $\mathbf{s}_j$). We know that $s_{j,i'} = s_{i,i'} \oplus (\mathbf{y}_{i,j})_{i'}$. Therefore we know that if $(\mathbf{y}_{i,j})_{i'} = 0$ then $s_{j,i'} = s_{i,i'}$ and if $(\mathbf{y}_{i,j})_{i'} = 1$ then $s_{j,i'} = 1 - s_{i,i'}$. We can thus replace the $\alpha_{j,i'}^{s_{j,i'}}$ element in the description of the function f with either $\alpha_{j,i'}^{s_{i,i'}}$ or $\alpha_{j,i'}^{1-s_{i,i'}}$, depending on the (known) value of $(\mathbf{y}_{i,j})_{i'}$. Doing this one variable at a time, results in an affine function of only $\mathbf{s}_i$. We again stress that we only used $\{\mathbf{y}_{i,j}\}_{i,j}$ for this transformation.

The challenger in this hybrid answers with $(c_0, \mathbf{c}) = (\beta' \cdot g_{0,i}^r, \boldsymbol{\alpha}' \cdot \mathbf{g}_i^r)$ instead of $(c_0, \mathbf{c}) = (\beta' \cdot \prod_{j\in[\ell]} {\alpha'}_j^{s_{i,j}} \cdot g_{0,i}^r, \mathbf{g}_i^r)$.

It holds that

$$\left|\Pr_{H_1}[b' = 1] - \Pr_{H_0}[b' = 1]\right| \leq 4nt\ell \cdot \mathrm{SGAdv}[\mathcal{B}_1] + O(nt\ell/T) \ ,$$

since, as in the proof of Theorem B.2, the difference between the hybrids can be viewed as a t-round interactive $(n\ell)$-vector game, considering $(\mathbf{g}_1, \ldots, \mathbf{g}_n)$ as the first message, and now the simulated adversary only needs a part of the answer for each query (the one that is respective to the user i for which the query was made).

Hybrid H_2. Following the proof outline of Theorem B.2, we change the distributions of $g_{0,i}$ for *all* $i \in [n]$, to $U(\mathbb{G}_L)$. The challenger still needs to know $\{\mathbf{y}_{i,j}\}_{i,j\in[n]}$ in order to answer the queries so we need to prove that even fixing $\{\mathbf{y}_{i,j}\}_{i,j\in[n]}$, hybrid H_2 is close to H_1. Intuitively speaking, this will require us to "extract" n uniform elements of $\mathbb{G}_L$ out of a single $\mathbf{s}$ (because once one of them is specified, all others are determined by the values of $\mathbf{y}_{i,j}$). Therefore, for security to hold, we have to require that ℓ is proportional to n.

We now wish to apply Lemma 2.2 to claim that the hybrids are statistically close. To do that, we consider the following family of hash functions (defined for a fixed value of $\{\mathbf{y}_{i,j}\}$)

$$z_{\mathbf{g}_1,\ldots,\mathbf{g}_n}(\mathbf{s}_1) = \left(\prod g_{1,i}^{s_{1,i}}, \prod g_{2,i}^{s_{1,i}\oplus(\mathbf{y}_{1,2})_i}, \ldots, \prod g_{n,i}^{s_{1,i}\oplus(\mathbf{y}_{1,n})_i}\right) \ .$$

This family is 2-universal (by a similar argument to Lemma 2.4, using the fact that the vectors $\mathbf{g}_i$ are independent). The output describes the distribution of $g_{0,i}$ in the case where all $\mathbf{y}_{i,j}$ are known, but $\mathbf{s}_1$ is not. We can now apply Lemma 2.2 respective to this family and conclude that

$$\left|\Pr_{H_2}[b' = 1] - \Pr_{H_1}[b' = 1]\right| \leq \sqrt{\frac{L^n}{2^{\ell+2}}} = \frac{1}{2} \cdot (L \cdot 2^{-\ell/n})^{n/2} \ .$$

Hybrid H_3. Note that at this point the public keys are distributed uniformly and independently of $\mathbf{y}_{i,j}$'s. We again change the way the challenger answers queries, along the lines of the proof of

Theorem B.2. Instead of answering with $(c_0, \mathbf{c}) = (\beta' \cdot g_{0,i}^r, \boldsymbol{\alpha}' \cdot \mathbf{g}_i^r)$, the challenger now answers with $(c_0, \mathbf{c}) = (g_{0,i}^r, \mathbf{g}_i^r)$. This can be viewed as a t-round interactive $n(\ell+1)$-vector game (similarly to the previous hybrid) and thus

$$\left| \Pr_{H_3}[b' = 1] - \Pr_{H_2}[b' = 1] \right| \leq 4nt(\ell+1) \cdot \mathrm{SGAdv}[\mathcal{B}_2] + O(nt\ell/T) \ .$$

Hybrid H_4. We revert the distributions of the $g_{0,i}$'s to the original one. As in hybrid H_2 we have $|\Pr_{H_4}[b' = 1] - \Pr_{H_3}[b' = 1]| \leq \frac{1}{2} \cdot (L \cdot 2^{-\ell/n})^{n/2}$.

Hybrid H_4 is identical to the $\mathrm{KDM}^{(n)}$ game with $b = 1$ as all queries are answered by encryptions of 0 and the claim follows. □

B.3.4 Beyond Affine Functions

Two building blocks have been suggested in [BGK09, BHHI09] to obtain KDM-security w.r.t. a larger family of functions. In this section we show that both of them can be based on the SG assumption.

B.3.4.1 Entropy-κ Security. The notion of entropy-κ KDM-security was introduced in [BGK09] as a way to extend KDM-security beyond affine functions. In their work [BGK09, Definitions 3.1, 3.2], an encryption scheme is called *projective* if the key-generation can be described as follows: first the secret key is uniformly sampled from some set $\mathcal{S}$, and then the public key is computed as a (possibly randomized) efficient function of the secret key. For $n \in \mathbb{N}$, a projective scheme is *entropy-κ* $\mathrm{KDM}^{(n)}$*-secure* if for any distribution $\mathcal{D}$ with $\mathbf{H}_\infty(\mathcal{D}) \geq \kappa$ supported inside $\mathcal{S}$, the scheme obtained by sampling the secret key from $\mathcal{D}$ rather than from $\mathcal{S}$ is $\mathrm{KDM}^{(n)}$-secure.

We show that $\mathcal{E}[\mathbb{G}_U, \ell]$, which is clearly projective, is entropy-κ $\mathrm{KDM}^{(1)}$-secure for $\kappa \geq \log L + \omega(\log k)$.

Lemma B.4. *Let $\mathcal{D}$ be a distribution on $\{0,1\}^\ell$ with $\mathbf{H}_\infty(\mathcal{D}) \geq \kappa$, let $\mathcal{E}_\mathcal{D}[\mathbb{G}_U, \ell]$ denote the encryption scheme that samples the secret key from $\mathcal{D}$ rather than $U(\{0,1\}^\ell)$.*

Let $\mathcal{A}$ be a $\mathrm{KDM}^{(1)}$-adversary for $\mathcal{E}_\mathcal{D}[\mathbb{G}_U, \ell]$ that makes at most t queries, then there exists an adversary $\mathcal{B}$ such that

$$\mathrm{KDM}^{(1)}\mathrm{Adv}[\mathcal{A}] \leq t(2\ell+1) \cdot \mathrm{SGAdv}[\mathcal{B}] + \sqrt{L \cdot 2^{-\kappa}} + O(t\ell/T) \ .$$

Proof. The proof is identical to that of Theorem B.2. The only difference is in the transitions from hybrid H_1 to H_2 and from hybrid H_3 to H_4 (which are the same transition in reverse order). The difference here is that now when we invoke Lemma 2.2, we have $|X| = 2^\kappa$ rather than 2^ℓ. The result immediately follows. □

The following corollary combines the above lemma with [BGK09, 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 B.5. *Consider $\mathcal{E}[\mathbb{G}_U, \ell]$ and let κ be polynomial in the security parameter such that $\kappa \geq \log L + \omega(\log k)$. Then for any entropy preserving set $\mathcal{H} = \{h_1, \ldots, h_\ell : h_i : \{0,1\}^\kappa \to \{0,1\}\}$*

of efficiently computable functions with polynomial cardinality (in the security parameter), there exists a $\mathrm{KDM}^{(1)}$*-secure scheme under the* SG*-assumption w.r.t. the class of functions*

$$\mathcal{F} = \left\{ f(\mathbf{x}) = \beta \cdot \prod_{i \in [\ell]} \alpha_i^{h_i(\mathbf{x})} : (\boldsymbol{\alpha}, \beta) \in {\mathbb{G}_M}^{\ell} \times \mathbb{G}_M \right\} .$$

B.3.4.2 Targeted Encryption. The notion of targeted encryption was introduced in [BHHI09] as a building block towards extending KDM security beyond affine functions. A targeted encryption scheme [BHHI09, Definition 3.1] consists of the following algorithms.

- *Key generation.* Takes the security parameter 1^k as input and outputs a key-pair (pk, sk) such that $sk = \mathbf{s} \in \{0,1\}^{\ell}$.
- *Targeted encryption.* Takes a public key pk, an index $i \in [\ell]$, a bit $b \in \{0,1\}$ and a message $m \in \mathcal{M}$ (for some message space $\mathcal{M}$) as input. Outputs a ciphertext c.
- *Targeted decryption.* Takes a secret key sk and a ciphertext c and outputs $m' \in \mathcal{M}$.

The following properties are required.

- *Targeted decryption.* For all $m \in \mathcal{M}$, $i \in [\ell]$ it holds that when generating a key pair (pk, sk), computing c by running the targeted encryption algorithm on (pk, i, s_i, m), where s_i is the i^{th} bit of the secret key, and then computing m' by running the decryption algorithm on (sk, c), it holds that $m' = m$.
- *Security against receiver.* For all $m_1, m_2 \in \mathcal{M}$, $i \in [\ell]$, consider c_1, c_2 obtained by generating a key pair (pk, sk) and then running the targeted encryption algorithm on $(pk, i, 1 - s_i, m_1), (pk, i, 1 - s_i, m_2)$, respectively. Then (sk, pk, c_1) and (sk, pk, c_2) are computationally indistinguishable.[14]
- *Security against outsiders.* For all $m \in \mathcal{M}$, $i \in [\ell]$, $b \in \{0,1\}$, consider c_1, c_2 obtained by generating a key pair (pk, sk) and then running the targeted encryption algorithm on $(pk, i, b, m_1), (pk, i, b, m_2)$, respectively. Then (pk, c_1) and (pk, c_2) are computationally indistinguishable.

An SG**-based targeted encryption scheme.** We now show how a slight modification of $\mathcal{E}[\mathbb{G}_U, \ell]$ provides a targeted encryption scheme. Our construction follows the general outline provided in [BHHI09] for converting $\mathrm{KDM}^{(1)}$-security w.r.t. affine functions into targeted encryption. Consider the scheme $\mathcal{T}[\mathbb{G}_U, \ell]$ presented below.

- *Parameters.* The parameters $\mathbb{G}_U, \ell$ have the same meaning as in $\mathcal{E}[\mathbb{G}_U, \ell]$. The message space is $\mathbb{G}_M$.
- *Key generation.* Identical to the key-generation of $\mathcal{E}[\mathbb{G}_U, \ell]$.
- *Targeted encryption.* On input $pk = (g_0, \mathbf{g})$, $i \in [\ell]$, $b \in \{0,1\}$, $m \in \mathbb{G}_M$, the encryption algorithm samples $r \xleftarrow{\$} [T^2]$, $u \xleftarrow{\$} \mathbb{G}_M$ and outputs $(c_0, \mathbf{c}) = (m \cdot u^{-b} \cdot g_0^r, u^{\mathbf{e_i}} \cdot \mathbf{g}^r)$.

[14] The definition in [BHHI09, Definition 3.1] is stricter and requires statistical indistinguishability, but as they mention, the one we provide here is sufficient to imply all of their results.

- *Targeted decryption.* On input $sk = \mathbf{s}$ and $(c_0, \mathbf{c})$, the decryption algorithm outputs $m' = c_0 \cdot \prod_{i\in[\ell]} c_i^{s_i}$.

The targeted decryption (completeness) property of $\mathcal{T}[\mathbb{G}_U, \ell]$ follows immediately by definition. The following two lemmas establish the security against receiver and security against outsiders properties for $\ell \geq \log L + \omega(\log k)$, based on the SG assumption.

Lemma B.6 (security against receiver)**.** *Let $i \in [\ell]$, $m \in \mathcal{M}$. Let $pk = (g_0, \mathbf{g}), sk = \mathbf{s}$ be a properly generated key-pair for $\mathcal{T}[\mathbb{G}_U, \ell]$ and let $r \xleftarrow{\$} [T^2]$, $u \xleftarrow{\$} \mathbb{G}_M$. Consider a distinguisher $\mathcal{A}$ between the distributions*

$$(\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-s_i)} \cdot g_0^r, u^{\mathbf{e_i}} \cdot \mathbf{g}^r)) \text{ and } (\mathbf{s}, (g_0, \mathbf{g}), (u \cdot g_0^r, \mathbf{g}^r)) \ ,$$

then there exists a SG *adversary $\mathcal{B}$ such that*

$$\mathrm{DistAdv}[\mathcal{A}] \leq 4\ell \cdot \mathrm{SGAdv}[\mathcal{B}] + O(\ell/T) \ .$$

Security against receiver follows since the latter distribution does not depend on m.

Proof. We note that the latter distribution $(\mathbf{s}, (g_0, \mathbf{g}), (u \cdot g_0^r, \mathbf{g}^r))$ is identical to $(\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-2s_i)} \cdot g_0^r, \mathbf{g}^r))$, since $u^{-(1-2s_i)}$ is uniformly distributed and independent of all other variables (using the fact that $s_i \in \{0, 1\}$) and thus $m \cdot u^{-(1-2s_i)}$ is uniform and independent of m. We thus consider a distinguisher $\mathcal{A}$ between $(\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-s_i)} \cdot g_0^r, u^{\mathbf{e_i}} \cdot \mathbf{g}^r))$ and $(\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-2s_i)} \cdot g_0^r, \mathbf{g}^r))$.

Consider the (efficiently computable) function

$$f(\mathbf{s}, m, u, \mathbf{g}, \mathbf{c}) = (\mathbf{s}, (\prod_{i\in[\ell]} g_i^{-s_i}, \mathbf{g}), (m \cdot u^{-(1-2s_i)} \cdot \prod_{i\in[\ell]} c_i^{-s_i}, \mathbf{c})) \ ,$$

and note that

$$\begin{aligned} f(\mathbf{s}, m, u, \mathbf{g}, \mathbf{g}^r) &= (\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-2s_i)} \cdot g_0^r, \mathbf{g}^r)) \\ f(\mathbf{s}, m, u, \mathbf{g}, u^{\mathbf{e_i}} \cdot \mathbf{g}^r) &= (\mathbf{s}, (g_0, \mathbf{g}), (m \cdot u^{-(1-s_i)} \cdot g_0^r, u^{\mathbf{e_i}} \cdot \mathbf{g}^r)) \ . \end{aligned}$$

Thus the adversary $\mathcal{A}$ can be used to distinguish between $(u, \mathbf{g}, u^{\mathbf{e_i}} \cdot \mathbf{g}^r)$ and $(u, \mathbf{g}, \mathbf{g}^r)$. It follows that $\mathrm{DistAdv}[\mathcal{A}] \leq 4\ell \cdot \mathrm{SGAdv}[\mathcal{B}] + O(\ell/T)$, since this is exactly a 1-round interactive ℓ-vector game (see Lemma B.1). □

Lemma B.7 (security against outsiders)**.** *Let $i \in [\ell]$, $b \in \{0, 1\}$, $m \in \mathcal{M}$. Let $pk = (g_0, \mathbf{g}), sk = \mathbf{s}$ be a properly generated key-pair for $\mathcal{T}[\mathbb{G}_U, \ell]$ and let $r \xleftarrow{\$} [T^2]$, $u \xleftarrow{\$} \mathbb{G}_M$. Consider a distinguisher $\mathcal{A}$ between the distributions*

$$((g_0, \mathbf{g}), (m \cdot u^{-b} \cdot g_0^r, u^{\mathbf{e_i}} \cdot \mathbf{g}^r)) \text{ and } ((g_0, \mathbf{g}), (g_0^r, \mathbf{g}^r)) \ ,$$

then there exists an adversary $\mathcal{B}$ such that

$$\mathrm{DistAdv}[\mathcal{A}] \leq 4(\ell + 1) \cdot \mathrm{SGAdv}[\mathcal{B}] + \sqrt{L \cdot 2^{-\ell}} + O(\ell/T) \ .$$

Security against outsiders follows since the latter distribution is independent of m.

Proof. First we consider an adversary $\mathcal{A}'$ that distinguishes the above distributions where g_0 is uniform in $\mathbb{G}_L$. In such case the above is a 1-round interactive $(\ell+1)$-vector game and thus $\mathrm{DistAdv}[\mathcal{A}'] \leq 4(\ell+1) \cdot \mathrm{SGAdv}[\mathcal{B}] + O(\ell/T)$. Since the real distribution of g_0 is $\frac{\sqrt{L \cdot 2^{-\ell}}}{2}$-uniform, the result follows (see hybrid H_3 in the proof of Theorem B.2 for a detailed explanation). □

The following is a corollary of the above, combined with [BHHI09, Theorem 4.1].

Corollary B.8. *Based on the* SG *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).*

Our scheme can also be used to obtain "augmented targeted encryption" and derive results for $\mathrm{KDM}^{(n)}$-security, but since the details are very similar to the above, they are omitted.

B.4 Leakage Resiliency

We prove that the scheme $\mathcal{E}[\mathbb{G}_U, \ell]$ is resilient to a leakage of up of $\lambda = \ell - \log(ML) - \omega(\log k)$ bits. The result is formally stated below, for overview see Section 7.

Theorem B.9. *Let $\mathcal{A}$ be a λ-leakage adversary for $\mathcal{E}[\mathbb{G}_U, \ell]$. Then there exists an adversary $\mathcal{B}$ such that*

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

Proof. We prove by a series of hybrids (experiment). Each experiment represents a process with a single binary value (one can think of 1 as a "success" in the experiment).

Hybrid H_0. This hybrid describes the following experiment: a challenger flips a coin $b \xleftarrow{\$} \{0,1\}$ and simulates the λ-leakage game with $\mathcal{A}$. It returns 1 if and only if $b' = b$, where b' is the value returned by $\mathcal{A}$. By definition

$$\Big|\Pr[H_0 = 1] - \frac{1}{2}\Big| = \frac{\mathrm{Leak}_\lambda\mathrm{Adv}[\mathcal{A}]}{2} \ .$$

Hybrid H_1. We change the encryption algorithm. In this hybrid, we encrypt a message m by first computing $\mathbf{c} = \mathbf{g}^r$ and then using $\mathbf{s}$ to produce $c_0 = m_b \cdot \prod_{i\in[\ell]} c_i^{-s_i}$. The ciphertext distribution does not change and hence $\Pr[H_1 = 1] = \Pr[H_0 = 1]$.

Hybrid H_2. Again we change the encryption. This time the challenger samples $\boldsymbol{\sigma} \xleftarrow{\$} {\mathbb{G}_M}^\ell$ and uses $\mathbf{c} = \boldsymbol{\sigma} \cdot \mathbf{g}^r$ instead of $\mathbf{c} = \mathbf{g}^r$. Note that the difference between H_1 and H_2 is exactly a 1-round interactive ℓ-vector game and thus by Lemma B.1, there exists an adversary $\mathcal{B}$ such that

$$|\Pr[H_2 = 1] - \Pr[H_1 = 1]| \leq 4\ell \cdot \mathrm{SGAdv}[\mathcal{B}] + O(\ell/T) \ .$$

Hybrid H_3. We notice that in H_2, the distribution of c_0 is

$$c_0 = m_b \cdot \prod_{i\in[\ell]} \sigma_i^{-s_i} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r \ .$$

In hybrid H_3 we change this distribution. The challenger samples $u \xleftarrow{\$} \mathbb{G}_M$ and sets

$$c_0 = m_b \cdot u^{-1} \cdot \Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r \ .$$

To analyze this hybrid, we recall that $\prod_{i\in[\ell]} g_i^{-s_i} \in \mathbb{G}_L$ and use Lemma 2.4 and Lemma 2.3 to conclude that even for a given $\mathbf{g}$ it holds that $(\boldsymbol{\sigma}, \prod_{i\in[\ell]} \sigma_i^{s_i}, \prod_{i\in[\ell]} g_i^{-s_i}, f(\mathbf{s}))$ is $\frac{1}{2}\cdot\sqrt{LM\cdot 2^{\lambda-\ell}}$-close to $(\boldsymbol{\sigma}, u, \prod_{i\in[\ell]} g_i^{-s_i}, f(\mathbf{s}))$. It follows that

$$|\Pr[H_3=1]-\Pr[H_2=1]| \leq \frac{1}{2}\cdot\sqrt{LM\cdot 2^{\lambda-\ell}}\ .$$

Hybrid H_4. We further change c_0 and now set it to be

$$c_0 = u\cdot\Big(\prod_{i\in[\ell]} g_i^{-s_i}\Big)^r\ .$$

Since u is uniform, it is distributed identically to $m_b\cdot u^{-1}$ and thus $\Pr[H_4=1]=\Pr[H_3=1]$. In H_4, however, the ciphertext distribution is independent of b. Therefore $\Pr[H_4=1]=\frac{1}{2}$. Combining all of the above, the result follows. □

B.5 Auxiliary-Input Resiliency

We present the general result, showing the auxiliary-input security of $\mathcal{E}[\mathbb{G}_U,\ell]$, following the same line of proof as in Section 8. The main difference is that we use a *generalized* Goldreich-Levin theorem (Theorem A.1) instead of the classic Theorem 2.5. Since the parameters of the generalized version are slightly worse, this leads to slightly worse parameters in the auxiliary input resiliency.

The following lemma establishes weak auxiliary-input security.

Lemma B.10. *Let $\epsilon(\ell)=M^{-\omega(\log\ell)}$ and let $f:\{0,1\}^\ell\to\{0,1\}^*$ be any ϵ-weakly uninvertible function (more precisely, family of functions). Let $\mathcal{A}$ be an f-auxiliary input adversary for $\mathcal{E}[\mathbb{G}_U,\ell]$. Then there exists an adversary $\mathcal{B}$ such that*

$$\mathrm{Aux}_f^{\mathrm{weak}}\mathrm{Adv}[\mathcal{A}] \leq 8\ell\cdot\mathrm{SGAdv}[\mathcal{B}]+O(\ell/T)+\mathrm{negl}(k)\ .$$

We remark that when M is polynomial, as in the case of the QR-based scheme described in Section 8, it is sufficient to require that ϵ is negligible, and thus *any* function that is weakly uninvertible in polynomial time meets the requirement.

The proof (below) follows the same steps as the proof of Theorem B.9. The only change is that we use the (generalized) Goldreich-Levin theorem instead of the leftover hash lemma.

Proof. We use the exact same hybrids as in the proof of Theorem B.9 (with the exception that f is now an ϵ-weakly uninvertible function rather than a length bounded one). The exact same arguments imply that

$$\begin{aligned}
\Big|\Pr[H_0=1]-\frac{1}{2}\Big| &= \frac{\mathrm{Aux}_f\mathrm{Adv}[\mathcal{A}]}{2}\\
\Pr[H_1=1] &= \Pr[H_0=1]\\
|\Pr[H_2=1]-\Pr[H_1=1]| &\leq 4\ell\cdot\mathrm{SGAdv}[\mathcal{B}]+O(\ell/T)\\
\Pr[H_4=1] &= \Pr[H_3=1]\\
\Pr[H_4=1] &= \frac{1}{2}\ .
\end{aligned}$$

Therefore, it remains to prove a bound on $|\Pr[H_3 = 1] - \Pr[H_2 = 1]|$. Assume towards contradiction that there exists a polynomial $t(k)$ such that $|\Pr[H_3 = 1] - \Pr[H_2 = 1]| \geq 1/t(k)$.

Consider the function $f' : \{0,1\}^\ell \to \{0,1\}^*$ defined as follows: $f'(\mathbf{s})$ uses $sk = \mathbf{s}$ as a secret key for $\mathcal{E}[\mathbb{G}_U, \ell]$ and computes a corresponding pk. It then computes $y \leftarrow f(sk, pk)$ and outputs (y, pk). Since f is ϵ-weak uninvertible, it follows that for any adversary $\mathcal{C}$, $\Pr[\mathcal{C}(f'(\mathbf{s})) = \mathbf{s}] < \epsilon$, where the probability is over $\mathbf{s}$ and over the coin-tosses of f' and $\mathcal{C}$.

We notice that the hybrids H_2 and H_3 can be represented as an efficient (randomized) function of the distributions $(f'(\mathbf{s}), \boldsymbol{\tau}, \langle \boldsymbol{\tau}, \mathbf{s} \rangle)$ and $(f'(\mathbf{s}), \boldsymbol{\tau}, v)$ respectively, where $(\boldsymbol{\tau}, v)$ are the discrete logarithms of $(\boldsymbol{\sigma}, u)$, respectively. Namely $\sigma_i = h^{\tau_i}$ and $u = h^v$. Note that τ_i, v are uniform in $\mathbb{Z}_M$. Our assumption implies, therefore, that these distributions are distinguishable with advantage $1/t(k)$. In this case, it follows from Theorem A.1 that there exists a $\mathcal{C}$ whose running time is at most $\mathrm{poly}(\ell, t(k)) = \mathrm{poly}(k)$, such that $\Pr[\mathcal{C}(f'(\mathbf{s})) = \mathbf{s}] \geq t(k) \cdot M^{-(1+\log(8\ell t^2(k)))}/8 = M^{-O(\log k)} > \epsilon$. We reached a contradiction and the claim, therefore, follows. □

An immediate corollary (see also [DGK+10, Lemma 4]) enables us to state that $\mathcal{E}[\mathbb{G}_U, \ell]$ is $(M^{-\omega(\log \ell)}/L)$-auxiliary input resilient. Note that in order for such functions to even exist it must be that $\ell \geq \log L + (\log M) \cdot \omega(\log k)$, since any function on $\{0,1\}^\ell$ is trivially invertible with probability at least $2^{-\ell}$.

Corollary B.11. *Let $\epsilon(\ell) = \frac{M^{-\omega(\log \ell)}}{L}$ and let f be any ϵ-uninvertible function (more precisely, family of functions). Let $\mathcal{A}$ be an f-auxiliary input adversary for $\mathcal{E}[\mathbb{G}_U, \ell]$. Then there exists an adversary $\mathcal{B}$ such that*

$$\mathrm{Aux}_f\mathrm{Adv}[\mathcal{A}] \leq 8\ell \cdot \mathrm{SGAdv}[\mathcal{B}] + O(\ell/T) + \mathrm{negl}(k) \ .$$

Proof. Recall that the public key of $\mathcal{E}[\mathbb{G}_U, \ell]$ is $(g_0 = \prod g_i^{-s_i}, \mathbf{g})$. Since $\mathbf{g}$ does not depend on the secret key, it can be treated as a public parameter of the scheme and not as a part of the public key. Since $g_0 \in \mathbb{G}_L$, then any ϵ-uninvertible function is also (ϵ/L)-weakly uninvertible. The result follows. □

We can derive the following corollary, which is a restatement of Corollary 8.2.

Corollary B.12. *Assuming that a subgroup indistinguishability assumption holds, then for any constant $\delta > 0$ there exists a $2^{-\ell^\delta}$-auxiliary input resilient encryption scheme.*

Proof. This follows immediately from Corollary B.11 by using $\mathcal{E}[\mathbb{G}_U, (t \cdot \omega(\log k))^{1/\delta}]$, where $t = \log T$, with T being the upper bound on $M \cdot L$ (note that $t \leq \mathrm{poly}(k)$). □