Efficient Cryptosystems From 2^k -th Power Residue Symbols

Fabrice Benhamouda, Javier Herranz, Marc Joye, Benoît Libert

Efficient Cryptosystems From 2^k-th Power Residue Symbols*

Fabrice Benhamouda[1], Javier Herranz[2], Marc Joye[3], and Benoît Libert[4,**]

[1] ENS Paris, CNRS, INRIA, and PSL
45 rue d'Ulm, 75230 Paris Cedex 06, France
fabrice.benhamouda@ens.fr
[2] Universitat Politècnica de Catalunya,
Dept. Matemàtica Aplicada
c. Jordi Girona 1-3, 08034, Barcelona, Spain
javier.herranz@upc.edu
[3] Technicolor
175 S. San Antonio Rd, Los Altos, CA 94022, USA
marc.joye@technicolor.com
[4] ENS Lyon, Laboratoire d'Informatique du Parallélisme
46 Allée d'Italie, 69364 Lyon Cedex 07, France
benoit.libert@ens-lyon.fr

Abstract. Goldwasser and Micali (1984) highlighted the importance of randomizing the plaintext for public-key encryption and introduced the notion of semantic security. They also realized a cryptosystem meeting this security notion under the standard complexity assumption of deciding quadratic residuosity modulo a composite number. The Goldwasser-Micali cryptosystem is simple and elegant but is quite wasteful in bandwidth when encrypting large messages. A number of works followed to address this issue and proposed various modifications.

This paper revisits the original Goldwasser-Micali cryptosystem using 2^k-th power residue symbols. The so-obtained cryptosystems appear as a very natural generalization for $k \geq 2$ (the case $k = 1$ corresponds exactly to the Goldwasser-Micali cryptosystem). Advantageously, they are efficient in both bandwidth and speed; in particular, they allow for fast decryption. Further, the cryptosystems described in this paper inherit the useful features of the original cryptosystem (like its homomorphic property) and are shown to be secure under a similar complexity assumption. As a prominent application, this paper describes an efficient lossy trapdoor function based thereon.

Keywords: Public-key encryption, quadratic residuosity, Goldwasser-Micali cryptosystem, homomorphic encryption, standard model.

1 Introduction

Encryption is arguably one of the most fundamental cryptographic primitives. Although it seems an easy task to identify properties that a good encryption scheme must fulfill, it turns out that rigorously defining the right security notion is not trivial at all. Security is context sensitive. Merely requiring that the plaintext cannot be recovered from the ciphertext is not enough in most applications. One may require that the knowledge of some *a priori* information on the plaintext does not help the adversary to obtain any new information, that is, beyond what can be obtained from the *a priori* information. This intuition is formally captured by the notion of *semantic security*, introduced in a seminal paper by Goldwasser and Micali [GM84]. They also introduced the equivalent notion of *indistinguishability of encryptions*, which is usually easier to work with. Given the encryption

* A preliminary version of this paper appears in the proceedings of EUROCRYPT 2013.
** Part of this work was done while this author was with Technicolor, France.

of any two equal-length (distinct) plaintexts, an adversary should not be able to distinguish the corresponding ciphertexts.

Clearly, the latter notion is only achievable by probabilistic encryption schemes. One such cryptosystem was also presented in [GM84]. It achieves ciphertext indistinguishability under the *Quadratic Residuosity* (QR) assumption. Informally, this assumption says that it is infeasible to distinguish squares from non-squares in $\mathbb{J}_N$ (i.e., the set of elements in $\mathbb{Z}_N^*$ whose Jacobi symbol is +1) where $N = pq$ is an RSA-type modulus of unknown factorization.

The Goldwasser-Micali cryptosystem is simple and elegant. The public key comprises an RSA modulus $N = pq$ and a non-square $y \in \mathbb{J}_N$ while the private key is the secret factor p. The encryption of a bit $m \in \{0,1\}$ is given by $c = y^m x^2 \bmod N$ for a random $x \in \mathbb{Z}_N^*$. The message m is recovered using p, by checking whether c is a square: $m = 0$ if so, and $m = 1$ otherwise —observe that a non-square $y \in \mathbb{J}_N$ is also a non-square modulo p. The encryption of a bitstring $m = (m_{k-1}, \ldots, m_0)_2$, with $m_i \in \{0,1\}$, proceeds by forming the ciphertexts $c_i = y^{m_i} x^2 \bmod N$, for $0 \leq i \leq k-1$. The scheme is computationally efficient but somewhat wasteful in bandwidth as $k \cdot \log_2 N$ bits are needed to encrypt a k-bit message. Several proposals were made to address this issue.

A first attempt is due to Blum and Goldwasser [BG84]. They achieve a better ciphertext expansion: the ciphertext has the same length as the plaintext plus an integer of the size of the modulus. The scheme is proved semantically secure assuming the unpredictability of the output of the Blum-Blum-Shub's pseudo-random generator [BBS82, BBS86], which resides on the factorization hardness assumption. Details about this scheme can be found in [Gol04].

Another direction, put forward by Benaloh and Fischer [CF85, Ben87], is to use a k-bit prime r such that $r \mid p-1$, $r^2 \nmid p-1$ and $r \nmid q-1$. The scheme also requires $y \in \mathbb{Z}_N^*$ such that $y^{\phi(N)/r} \not\equiv 1 \pmod N$, where $\phi(N) = (p-1)(q-1)$ denotes Euler's totient function. A k-bit message m (with $m < r$) is encrypted as $c = y^m x^r \bmod N$, where $x \in_R \mathbb{Z}_N^*$. It is recovered by searching over the entire message space, $[0, r) \subseteq \{0,1\}^k$, for the element m satisfying $(y^{\phi(N)/r})^m \equiv c^{\phi(N)/r} \pmod N$. The scheme is shown to be secure under the *prime-residuosity assumption* (which generalizes the quadratic residuosity assumption). With the Benaloh-Fischer cryptosystem, the ciphertext corresponding to a k-bit message is short but the decryption process is now demanding. In practice, the scheme is therefore limited to small values of k, say $k < 40$.

The Benaloh-Fischer cryptosystem was subsequently extended by Naccache and Stern [NS98]. They observe that the decryption can be sped up by rather considering a product of small (odd) primes $R = \prod_i r_i$ such that $r_i \mid \phi(N)$ but ${r_i}^2 \nmid \phi(N)$ for each prime r_i. Given a ciphertext, the plaintext m is reconstructed from $m_i := m \bmod r_i$ through Chinese remaindering. The advantage is that each m_i is searched in the subspace $[0, r_i)$ instead of the entire message space. A variant of this technique was used by Groth [Gro05].

Other generalizations and extensions of the Goldwasser-Micali cryptosystem but without formal security analysis can be found in [ZMI88, KKOT90, PLW95]. In [MV04b, MV04a], Monnerat and Vaudenay developed applications using the more general theory of characters, specifically with characters of order ≤ 4. Related cryptosystems are described in [SW95, Sch98]. A different approach was proposed by Okamoto and Uchiyama [OU98], who suggested to use moduli of the form $N = p^2 q$. This allows encrypting messages of size up to $\log_2 p$ bits. This was later extended by Paillier [Pai99] to the setting $N = p^2 q^2$; see also [CGHGN01, DJN10].

A useful application of additive homomorphic encryption schemes resides in the construction of *lossy trapdoor functions* (or LTDFs in short). These functions, as introduced by Peikert and Waters [PW08], are function families wherein injective functions are computationally indistin-

guishable from *lossy* functions, which lose many bits of information about their input. LTDFs have proved to be very powerful and versatile in the cryptographer's toolbox. They notably imply chosen-ciphertext-secure public-key encryption [PW08], deterministic encryption [BBO07, BFO08], as well as cryptosystems that retain some security in the absence of reliable randomness [BBN+09] or in the presence of selective-opening adversaries [BHY09].

Our contributions

New Homomorphic Cryptosystem. We suggest an improvement of the original Goldwasser-Micali cryptosystem. It can be seen as a follow-up of the earlier works due to Benaloh and Fischer [CF85] and Naccache and Stern [NS98]. Before discussing it, we quote from [NS98]:

> *"Although the question of devising new public-key cryptosystems appears much more difficult* [...] *we feel that research in this direction is still in order*: *simple yet efficient constructions may have been overlooked."*

It is striking that the generalized cryptosystem in this paper was not already proposed because, as will become apparent (cf. Section 3), it turns out to be a very natural generalization. Our approach consists in considering n^{th}-power residues modulo N with $n = 2^k$ (the Goldwasser-Micali system corresponds to the case $k = 1$). This presents many advantages. First, the resulting cryptosystem is bandwidth-efficient. Only $\log_2 N$ bits are needed for encrypting a k-bit message in typical applications (e.g., using the KEM/DEM paradigm). Second, the decryption process is fast. Searches are no longer needed (not even in smaller subspaces) in the decryption algorithm as plaintext messages can be recovered bit by bit. Further, although asymptotically slower than in Paillier's cryptosystem, the decryption process turns out to achieve comparable performance for most practical values of k (e.g., $k \leq 128$). As a last advantage, the underlying complexity assumptions are similar to that used by Goldwasser and Micali. The proposed cryptosystem is shown to be secure under the quadratic residuosity assumption for RSA moduli $N = pq$ such that $p \equiv 1 \pmod{2^k}$ and $q \equiv 3 \pmod 4$. When $q \not\equiv 3 \pmod 4$, it assumes in addition the hardness of determining the Jacobi symbol of an element $y \in \mathbb{Z}_N^*$ given a pair (x, N) where $x = y^2 \bmod N$. Although the proposed cryptosystem makes use of primes of special form, there are no known factoring algorithms taking advantage of that. Further, complexity-wise, the use of such special primes does not incur penalty with the latest prime generation algorithms. As will be seen, the time required to generate a random prime $p \equiv 1 \pmod{2^k}$ is essentially the same as the time required to generate a random, form-free prime.

We also note that, similarly to the Goldwasser-Micali cryptosystem, our generalized cryptosystem enjoys an additive property known as *homomorphic encryption*. If c_1 and c_2 denote two ciphertexts corresponding to k-bit plaintexts m_1 and m_2, respectively, then $c_1 \cdot c_2 \pmod N$ is an encryption of the message $m_1 + m_2 \pmod{2^k}$. This reveals useful in several applications like voting schemes.

As another useful property, the new scheme inherits the selective opening security[5] [DNRS03, BHY09] of the Goldwasser-Micali system (in the sense of a simulation-based definition given in [BHY09]). We actually prove its semantic security by showing that its public key is indistinguishable from a so-called *lossy* key for which encryptions reveal nothing about the encrypted message.

[5] This notion refers to an attack scenario where the adversary is given t encryptions of possibly correlated messages, opens $t/2$ out of these (and thereby obtains the messages *and* encryption coins) before attempting to harm the security of the remaining ciphertexts.

We thus believe our system to provide an interesting competitor to Paillier's cryptosystem for certain applications. As a salient example, we show that it provides a dramatically improved lossy trapdoor function.

New Efficient Lossy Trapdoor Functions. The initial LTDF realizations [PW08] were based on the Decisional Diffie-Hellman (DDH) and Learning-with-Error (LWE) [Reg09] assumptions. More efficient examples based on the Decisional Composite Residuosity (DCR) assumption were given in [BFO08, FGK+10, FGK+13] while Kiltz *et al.* [KOS10] showed that the RSA permutation provides a lossy function. Under the Quadratic Residuosity (QR) assumption, three distinct constructions were put forth in [HO12, FGK+10, FGK+13, Wee12]. Those of Freeman *et al.* [FGK+10, FGK+13] and of Wee [Wee12] must be used in combination with the results of Mol and Yilek [MY10] as they only lose single bits of information about the input. Hemenway and Ostrovsky [HO12] suggested a more efficient realization, of which Wee's framework [Wee12] is a generalization. While their QR-based LTDF has found applications in the design of deterministic encryption schemes [BS11], it is conceptually very similar to the Peikert-Waters matrix-based schemes and suffers from similarly large outputs and descriptions.

We show that our variant of the Goldwasser-Micali cryptosystem drastically improves the efficiency of the Hemenway-Ostrovsky LTDF. Specifically, it reduces both the length of the output and the description of the function. By appropriately selecting the parameters, we obtain evaluation keys and outputs consisting of a constant number of $\mathbb{Z}_N^*$ elements. We thus get a DDH/QR-based LTDF, whose efficiency is competitive with Paillier-based realizations [BFO08, FGK+10, FGK+13]. These improvements carry over to the deterministic encryption setting, when the Hemenway-Ostrovsky LTDF is used as a building block of the Brakerski-Segev system [BS11].

Outline of the paper

In the next section, we introduce some mathematical background and review some complexity assumptions. In Section 3, we present our generalized cryptosystem. We prove its security in Section 4. Section 5 discusses certain implementation aspects. In Section 6, we describe our new lossy trapdoor function. Finally, we conclude in Section 7.

2 Background

We review some useful background and fix the notation. In particular, we define the n-th power residue symbol. We refer the reader to [IR90, Sho10, Yan02] for further details on (quadratic) residuosity. More information about encryption schemes can be found in textbooks in cryptography; e.g. [Gol04, KL07].

2.1 General notation

The set of non-negative integers is denoted by $\mathbb{N}$. For any integer $N \geq 2$, $\mathbb{Z}_N$ denotes the ring of integers modulo N, and $\mathbb{Z}_N^*$ denotes its group of units. The order of $\mathbb{Z}_N^*$ is $\phi(N)$, where ϕ is Euler's totient function.

For any positive integer N and any integer a, $a \bmod N$ represents the smallest integer in the set $\{0, \ldots, N-1\}$ that is congruent to a modulo N. Furthermore, for any positive odd integer N and any

integer a, a mods N represents the absolute smallest residue of a modulo N —note the "s" ending the "mod" operator. The complete set of absolute smallest residues is $\{-(N-1)/2, \ldots, -1, 0, 1, \ldots, (N-1)/2\}$.

2.2 n^{th}-power residues

Let $N \geq 2$ be an integer. For each integer $n \geq 2$, we define $(\mathbb{Z}_N^*)^n = \{x^n \mid x \in \mathbb{Z}_N^*\}$ as the set of n^{th}-*power residues modulo* N. If the relation $a = x^n$ has no solution in $\mathbb{Z}_N^*$ then a is called a n^{th}-*power non-residue modulo* N.

Suppose that p is an odd prime. For any integer a with $\gcd(a, p) = 1$, it is easily verified that a is a n^{th}-power residue modulo p if and only if

$$a^{\frac{p-1}{\gcd(n,p-1)}} \equiv 1 \pmod{p} \ .$$

When $n = 2$ (and so $\gcd(n, p-1) = 2$), this is known as Euler's criterion. It allows one to distinguish quadratic residues from quadratic non-residues. This defines the *Legendre symbol*:

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p \end{cases} .$$

There are several ways to generalize the Legendre symbol (see [Lem00]). In this paper, we consider the n-th power residue symbol for a divisor n of $(p-1)$, as presented in [Yan02, Definition 1.6.21].

Definition 1. *Let p be an odd prime and let $n \geq 2$ such that $n \mid p-1$. Then the symbol*

$$\left(\frac{a}{p}\right)_n = a^{\frac{p-1}{n}} \text{ mods } p$$

is called the n-th power residue symbol modulo p.

It satisfies the following properties. Let a and b be two integers that are co-prime to p. Then:

1. If $a \equiv b \pmod{p}$ then $\left(\frac{a}{p}\right)_n = \left(\frac{b}{p}\right)_n$;
2. $\left(\frac{a^n}{p}\right)_n = 1$;
3. $\left(\frac{ab}{p}\right)_n = \left(\frac{a}{p}\right)_n \left(\frac{b}{p}\right)_n$ mods p;
4. $\left(\frac{1}{p}\right)_n = 1$ and $\left(\frac{-1}{p}\right)_n = (-1)^{\frac{p-1}{n}}$.

2.3 Quadratic residuosity

Let $N = pq$ be the product of two (odd) primes p and q. For an integer a co-prime to N, the *Jacobi symbol* is the product of the corresponding Legendre symbols, namely $\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$. This gives rise to the multiplicative group $\mathbb{J}_N$ of integers whose Jacobi symbol is +1, $\mathbb{J}_N = \{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{N}\right) = 1\}$. A relevant subset of $\mathbb{J}_N$ is the set of quadratic residues modulo N, $\mathbb{QR}_N = \{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1\}$. The set of integers whose Jacobi symbol is -1 is denoted by $\bar{\mathbb{J}}_N$; i.e., $\bar{\mathbb{J}}_N = \{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{N}\right) = -1\} = \mathbb{Z}_N^* \setminus \mathbb{J}_N$.

The *Quadratic Residuosity* (QR) assumption says that, given a random element $a \in \mathbb{J}_N$, it is hard to decide whether $a \in \mathbb{QR}_N$ if the prime factors of N are unknown. To emphasize that this should hold for RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$ for some $k \geq 1$, we refer to it as the k-QR *assumption*. Formally, we have:

Definition 2 (Quadratic Residuosity Assumption, k-QR). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p \equiv 1 \pmod{2^k}$, and their product $N = pq$. The* Quadratic Residuosity (k-QR) *assumption asserts that the function* $\mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}}(\kappa)$*, defined as the distance*

$$\left|\Pr[\mathcal{D}(x, N) = 1 \mid x \xleftarrow{R} \mathbb{QR}_N] - \Pr[\mathcal{D}(x, N) = 1 \mid x \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of running $(N, p, q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ and choosing at random $x \in \mathbb{QR}_N$ and $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

We also introduce a new assumption. The new assumption, which we call the *Squared Jacobi Symbol* (SJS) assumption, posits the infeasibility of determining whether $\left(\frac{y}{N}\right) = 1$ or -1 given (x, N) where $x = y^2 \bmod N$. Again, when the assumption is directed to RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$, we write it k-SJS. Formally, we define:

Definition 3 (Squared Jacobi Symbol Assumption, k-SJS). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p \equiv 1 \pmod{2^k}$, and their product $N = pq$. The* Squared Jacobi Symbol (k-SJS) *assumption asserts that the function* $\mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}}(\kappa)$*, defined as the distance*

$$\left|\Pr[\mathcal{D}(y^2 \bmod N, N) = 1 \mid y \xleftarrow{R} \mathbb{J}_N] - \Pr[\mathcal{D}(y^2 \bmod N, N) = 1 \mid y \xleftarrow{R} \overline{\mathbb{J}}_N]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of running $(N, p, q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ and choosing at random $y \in \mathbb{J}_N$ and $y \in \overline{\mathbb{J}}_N$.

When $q \equiv 3 \pmod 4$, any element $x \in \mathbb{QR}_N$ has four square roots: two of Jacobi symbol +1 and two of Jacobi symbol −1. In that case, as detailed in Section 3.3, the k-SJS assumption holds perfectly.

3 A New Public-Key Encryption Scheme

We generalize the Goldwasser-Micali cryptosystem so that it can efficiently support the encryption of larger messages while remaining additively homomorphic.

3.1 Description

The setting is basically the same as for the Goldwasser-Micali cryptosystem. The only additional requirement is that the prime p is chosen congruent to 1 modulo 2^k, where k denotes the bit-size of the messages being encrypted. The case $k = 1$ (i.e., encryption of 1-bit messages) corresponds to the Goldwasser-Micali cryptosystem.

In more detail, our encryption scheme is the tuple (KeyGen, Encrypt, Decrypt) defined as follows.

KeyGen(1^κ) Given a security parameter κ, KeyGen defines an integer $k \geq 1$, randomly generates primes p and q such that $p \equiv 1 \pmod{2^k}$, and sets $N = pq$. It also picks a random $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$. The public and private keys are $pk = \{N, y, k\}$ and $sk = \{p\}$, respectively.

Encrypt(pk, m) Let $\mathcal{M} = \{0,1\}^k$. To encrypt a message $m \in \mathcal{M}$ (seen as an integer in $\{0, \ldots, 2^k - 1\}$), Encrypt picks a random $x \in \mathbb{Z}_N^*$ and returns the ciphertext $c = y^m \, x^{2^k} \bmod N$.

Decrypt(sk, c) Given $c \in \mathbb{Z}_N^*$ and the private key $sk = \{p\}$, the algorithm first computes $z = \left(\frac{c}{p}\right)_{2^k}$ and then finds $m \in \{0, \ldots, 2^k - 1\}$ such that the relation

$$z = \left[\left(\frac{y}{p}\right)_{2^k}\right]^m \text{ mods } p$$

holds. A fast decryption algorithm is detailed in Section 3.2.

The correctness of the decryption is easily verified by observing that $\alpha := \left(\frac{y}{p}\right)_{2^k}$ has order 2^k as an element in $\mathbb{Z}_p^*$. Indeed, letting $n = \mathrm{ord}_p(\alpha)$ the order of α, we have $n \mid 2^k$ since, by definition, $\alpha \equiv y^{\frac{p-1}{2^k}} \pmod{p}$. But n cannot be equal to $2^{k'}$ for some $k' < k$ because $\alpha^{2^{k'}} \equiv 1 \pmod{p}$ would imply $y^{\frac{p-1}{2}} \equiv 1 \pmod{p}$, which contradicts the assumption that $y \in \mathbb{J}_N \setminus \mathbb{QR}_N \iff \left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. The decryption algorithm recovers the unique $m \in \{0, \ldots, 2^k - 1\}$ such that $\alpha^m \equiv z \pmod{p}$.

Furthermore, the scheme is homomorphic for the addition modulo 2^k: if $c_1 = y^{m_1} \, {x_1}^{2^k}$ and $c_2 = y^{m_2} \, {x_2}^{2^k}$ are ciphertexts of m_1 and m_2 respectively, then $c_1 \cdot c_2 = y^{m_1+m_2} \, (x_1 x_2)^{2^k} \bmod N$ is a ciphertext of $m_1 + m_2 \pmod{2^k}$.

3.2 Fast decryption

At first glance, from the above description, it seems that the decryption process amounts to a search through the entire message space $\{0,1\}^k$, similarly to some earlier cryptosystems. But we can do better. One of the main advantages of the proposed cryptosystem is that it provides an efficient way to recover the message. Hence, it remains practical, even for large values of k. The decryption algorithm proceeds similarly to the Pohlig-Hellman algorithm [PH78].

The message $m \in \{0,1\}^k$ is viewed as a k-bit integer given by its binary expansion $m = \sum_{i=0}^{k-1} m_i \, 2^i$, with $m_i \in \{0,1\}$. Given $c = y^m x^{2^k} \bmod N$, we have

$$\left(\frac{c}{p}\right)_{2^i} = \left(\frac{y^m x^{2^k}}{p}\right)_{2^i} = \left(\frac{y^{\sum_{j=0}^{i-1} m_j \, 2^j}}{p}\right)_{2^i} = \left(\frac{y}{p}\right)_{2^i}^{\sum_{j=0}^{i-1} m_j \, 2^j} \text{ mods } p$$

since $y^m x^{2^k} = y^{\sum_{j=0}^{i-1} m_j \, 2^j} \cdot \left(y^{\sum_{j=i}^{k-1} m_j \, 2^{j-i}} x^{2^{k-i}}\right)^{2^i}$, for $1 \leq i \leq k$. As a result, m can be recovered bit by bit using p, starting from the least significant bit. Implementation details are provided in Section 5.2.

3.3 Security analysis

We focus on semantic security. The case $k = 1$ corresponds to the Goldwasser-Micali cryptosystem. Indeed, when $k = 1$, the 2^k-th power residue symbol is then the classical Legendre symbol

and the assumption $p \equiv 1 \pmod{2^k}$ is trivially verified. The Goldwasser-Micali scheme has indistinguishable encryptions under the standard Quadratic Residuosity assumption.

In the general case (i.e., $k \geq 1$), we prove that the scheme provides indistinguishable encryptions (IND-CPA security) under the k-QR and k-SJS assumptions. More precisely:

Theorem 1. *Let κ denote the security parameter. For any* IND-CPA *adversary $\mathcal{A}$ against the scheme of Section 3.1, there exist a k-*QR *distinguisher $\mathcal{D}_1$ and a k-*SJS *distinguisher $\mathcal{D}_2$ with comparable running times and such that*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{A}}(\kappa) \leq \tfrac{3}{2}\left((k-\tfrac{1}{3})\cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}_1}(\kappa) + (k-1)\cdot \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}_2}(\kappa)\right) \ .$$

Proof. The proof is given in Section 4. □

When $k = 1$, the theorem reads $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{A}}(\kappa) \leq \mathbf{Adv}^{\text{QR}}_{\mathcal{D}_1}(\kappa)$, as shown in [GM84].

We henceforth assume $k \geq 2$. When $k \geq 2$, the condition $p \equiv 1 \pmod{2^k}$ implies $p \equiv 1 \pmod 4$. Depending on q, there are two possible sub-cases. If $q \equiv 1 \pmod 4$ then -1 is a square modulo p and modulo q. The square roots of any element of $\mathbb{QR}_N$ then all have the same Jacobi symbol modulo N. The hardness to distinguish among them is captured by the k-SJS assumption.

The sub-case $q \equiv 3 \pmod 4$ is more interesting. We then have $\left(\frac{-1}{N}\right) = -1$. As a consequence, by definition of the Jacobi symbol, it follows that

$$\begin{aligned}\left\{y^2 \bmod N \mid y \in \bar{\mathbb{J}}_N\right\} &= \left\{y^2 \bmod N \mid \left(\tfrac{y}{N}\right) = -1\right\} = \left\{(-y)^2 \bmod N \mid \left(\tfrac{-y}{N}\right) = -1\right\} \\ &= \left\{y^2 \bmod N \mid -\left(\tfrac{y}{N}\right) = -1\right\} \\ &= \left\{y^2 \bmod N \mid y \in \mathbb{J}_N\right\} \ .\end{aligned}$$

Since the two sets are identical, the k-SJS assumption holds perfectly when $q \equiv 3 \pmod 4$. This in turn leads to the following corollary.

Corollary 1. *When $q \equiv 3 \pmod 4$, for any* IND-CPA *adversary $\mathcal{A}$ against the scheme of Section 3.1, there exists a k-*QR *distinguisher $\mathcal{D}$ with comparable running time and such that*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{A}}(\kappa) \leq \tfrac{1}{2}(3k-1)\cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}}(\kappa) \ .$$

Proof. First observe that the bound is valid for $k = 1$. For $k \geq 2$, the corollary follows by letting $\mathcal{D}_1 = \mathcal{D}$ and plugging $\mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}_2}(\kappa) = 0$ in the bound of Theorem 1. □

The bound in Corollary 1 can be slightly tightened by a more direct proof. We have:

Theorem 2. *Let κ denote the security parameter. For any* IND-CPA *adversary $\mathcal{A}$ against the scheme of Section 3.1 with $q \equiv 3 \pmod 4$, there exists a k-*QR *distinguisher $\mathcal{D}$ with comparable running time and such that*

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{A}}(\kappa) \leq \tfrac{1}{2}(k+1)\cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}}(\kappa) \ .$$

Proof. The proof is given in appendix. □

Comparing the security bounds offered by Theorems 1 and 2, it turns out that RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$ and $q \equiv 3 \pmod 4$ should be preferred over RSA moduli with $q \equiv 1 \pmod 4$. More importantly, selecting RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$ and $q \equiv 3 \pmod 4$ presents the advantage that the security solely relies on a QR-based assumption (namely, the k-QR assumption).

Regarding the weaker notion of one-wayness, it is easy to see that one-wayness can be proved just under the k-QR assumption in all cases. Let $\mathcal{B}$ be an adversary which returns m when given $c = y^m x^{2^k} \bmod N$ and N (with $x \xleftarrow{R} \mathbb{Z}_N^*$). We construct a distinguisher $\mathcal{D}$ for the k-QR assumption as follows. It takes as input an RSA modulus $N = pq$ with $p \equiv 1 \pmod{2^k}$ and an element $w \in \mathbb{Z}_N^*$. Its goal is to distinguish whether $w \in \mathbb{QR}_N$ or $w \in \mathbb{J}_N \setminus \mathbb{QR}_N$. To do this, $\mathcal{D}$ simply picks a random $x \in \mathbb{Z}_N^*$, sets $c = wx^2 \bmod N$, and feeds $\mathcal{B}$ with (c, N). When the latter outputs a result m, $\mathcal{D}$ outputs the least significant bit of m. It is clear that if $w \in \mathbb{QR}_N$, c is a ciphertext of an even plaintext; otherwise, c is a ciphertext of an odd plaintext. Hence if $\mathcal{B}$ is a successful attacker against one-wayness, $\mathcal{D}$ is a successful distinguisher for k-QR.

4 Security Proof

4.1 Gap 2^k-residuosity assumption

The k-QR assumption states that, without knowing the factorization of N, random elements of $\mathbb{QR}_N$ are computationally indistinguishable from random elements of $\mathbb{J}_N \setminus \mathbb{QR}_N$. Here, it will be convenient to consider a *gap* variant of the k-QR assumption. We chose the terminology "gap" (not to be confused with computational problems which have an easy decisional counterpart [OP01]) by analogy with certain lattice problems, where not every instance is a YES or NO instance since a gap exists between these.

Definition 4 (Gap 2^k-Residuosity Assumption, Gap 2^k-Res). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p \equiv 1 \pmod{2^k}$. The Gap 2^k-Residuosity problem in $\mathbb{Z}_N^*$ consists in distinguishing a uniform element of V_0 from a uniform element of V_1 given only $N = pq$, where V_0 and V_1 are defined as follows:*

$$V_0 = \{x \in \mathbb{J}_N \setminus \mathbb{QR}_N\} \quad \textit{and} \quad V_1 = \{y^{2^k} \bmod N \mid y \in \mathbb{Z}_N^*\} \ .$$

The Gap 2^k-Residuosity (Gap 2^k-Res) *assumption posits that the advantage* $\mathbf{Adv}_{\mathcal{D}}^{\text{Gap } 2^k\text{-Res}}(\kappa)$, *defined as the distance*

$$\left|\Pr[\mathcal{D}(x,k,N) = 1 \mid x \xleftarrow{R} V_0] - \Pr[\mathcal{D}(x,k,N) = 1 \mid x \xleftarrow{R} V_1]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of running $(N, p, q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ and choosing $x \xleftarrow{R} V_0$ and $x \xleftarrow{R} V_1$.

The latter assumption was independently considered in [ABP13] by Abdalla, Ben Hamouda and Pointcheval who used it to provide tighter security proofs for forward-secure signatures.

4.2 Gap 2^k-Res is implied by k-QR and k-SJS

We now investigate the relationship between the Gap 2^k-Residuosity assumption and other more natural assumptions; namely, we will show that Gap 2^k-Res is implied by the k-QR and k-SJS assumptions.

For this proof, it is useful to introduce two intermediate assumptions: the "special" k-QR assumption and the "special" k-SJS assumption.

Definition 5 (Special Quadratic Residuosity Assumption, k-QR*). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p \equiv 1 \pmod{2^k}$, and their product $N = pq$. The* Special Quadratic Residuosity (k-QR*) *assumption asserts that the function* $\mathbf{Adv}^{k\text{-QR}^\star}_{\mathcal{D}}(\kappa)$*, defined as the distance*

$$\left|\Pr[\mathcal{D}(x,N)=1 \mid x = y^2 \bmod N, y \xleftarrow{R} \mathbb{J}_N] - \Pr[\mathcal{D}(x,N)=1 \mid x \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of running $(N,p,q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ and choosing at random $y \in \mathbb{J}_N$ and $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

Definition 6 (Special Squared Jacobi Symbol Assumption, k-SJS*). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p \equiv 1 \pmod{2^k}$, and their product $N = pq$. The* Special Squared Jacobi Symbol (k-SJS*) *assumption asserts that the function* $\mathbf{Adv}^{k\text{-SJS}^\star}_{\mathcal{D}}(\kappa)$*, defined as the distance*

$$\left|\Pr[\mathcal{D}(y^2 \bmod N, N)=1 \mid y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N] - \Pr[\mathcal{D}(y^2 \bmod N, N)=1 \mid y \xleftarrow{R} \bar{\mathbb{J}}_N]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of running $(N,p,q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ and choosing at random $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$ and $y \in \bar{\mathbb{J}}_N$.

Lemma 1. *Using the previous notation, we have k-QR + k-SJS $\implies$ k-QR* + k-SJS*. More precisely, for any probabilistic polynomial-time distinguisher $\mathcal{A}$ against k-QR* or k-SJS*, $\mathcal{A}$ is also a distinguisher against k-QR or k-SJS and there exists a distinguisher $\mathcal{B}$ against k-QR with comparable running time, such that*

$$\mathbf{Adv}^{k\text{-QR}^\star}_{\mathcal{A}}(\kappa) \le \mathbf{Adv}^{k\text{-QR}}_{\mathcal{A}}(\kappa) + \tfrac{1}{2}\,\mathbf{Adv}^{k\text{-SJS}}_{\mathcal{A}}(\kappa)\,,$$
$$\mathbf{Adv}^{k\text{-SJS}^\star}_{\mathcal{A}}(\kappa) \le \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{A}}(\kappa) + \tfrac{1}{2}\,\mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}}(\kappa)\,.$$

Proof. Consider a probabilistic polynomial-time algorithm $\mathcal{A}$ taking as input N and $x \in \mathbb{J}_N$. For $x \xleftarrow{R} \mathbb{J}_N$, we let

$$\begin{cases} \epsilon_1 = \Pr[\mathcal{A}(x,N)=1 \mid x \in \mathbb{J}_N \setminus \mathbb{QR}_N] \\ \epsilon'_2 = \Pr[\mathcal{A}(x,N)=1 \mid x = y^2 \in \mathbb{QR}_N \wedge y \in \mathbb{J}_N \setminus \mathbb{QR}_N] \\ \epsilon''_2 = \Pr[\mathcal{A}(x,N)=1 \mid x = y^2 \in \mathbb{QR}_N \wedge y \in \mathbb{QR}_N] \\ \epsilon_3 = \Pr[\mathcal{A}(x,N)=1 \mid x = y^2 \in \mathbb{QR}_N \wedge y \in \bar{\mathbb{J}}_N] \end{cases}.$$

Against k-QR, k-SJS, k-QR*, and k-SJS*, its advantage is denoted

$$\alpha_1 := \left|\epsilon_1 - \tfrac{1}{4}(\epsilon'_2 + \epsilon''_2) - \tfrac{1}{2}\epsilon_3\right|\,,\quad \alpha_2 := \left|\tfrac{1}{2}(\epsilon'_2 + \epsilon''_2) - \epsilon_3\right|\,,\quad \alpha_3 := \left|\epsilon_1 - \tfrac{1}{2}(\epsilon'_2 + \epsilon''_2)\right|\,,\quad \alpha_4 := \left|\epsilon'_2 - \epsilon_3\right|\,,$$

respectively.

We have to show that if the k-QR and k-SJS assumptions hold then so do the k-QR* and k-SJS* assumptions. The k-QR and k-SJS assumptions imply that α_1 and α_2 are negligible. We also note that any significant difference between ϵ'_2 and ϵ''_2 would lead to a distinguisher against k-QR. We thus have $|\epsilon'_2 - \epsilon''_2| \leq \mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}}(\kappa)$, with $\mathcal{B}$ an algorithm with running time comparable to that of $\mathcal{A}$.

From the definitions of α_3 and α_4, we can write

$$\begin{aligned}\alpha_3 &= \left|\epsilon_1 - \tfrac{1}{2}(\epsilon'_2 + \epsilon''_2)\right| = \left|\epsilon_1 - \tfrac{1}{4}(\epsilon'_2 + \epsilon''_2) - \tfrac{1}{2}\epsilon_3 + \tfrac{1}{2}\epsilon_3 - \tfrac{1}{4}(\epsilon'_2 + \epsilon''_2)\right| \\ &\leq \left|\epsilon_1 - \tfrac{1}{4}(\epsilon'_2 + \epsilon''_2) - \tfrac{1}{2}\epsilon_3\right| + \left|\tfrac{1}{2}\epsilon_3 - \tfrac{1}{4}(\epsilon'_2 + \epsilon''_2)\right| \\ &= \alpha_1 + \tfrac{1}{2}\mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}}(\kappa)\end{aligned}$$

and

$$\begin{aligned}\alpha_4 &= \left|\epsilon'_2 - \epsilon_3\right| = \left|\tfrac{1}{2}\epsilon'_2 + \tfrac{1}{2}\epsilon''_2 - \epsilon_3 + \tfrac{1}{2}\epsilon'_2 - \tfrac{1}{2}\epsilon''_2\right| \leq \left|\tfrac{1}{2}(\epsilon'_2 + \epsilon''_2) - \epsilon_3\right| + \left|\tfrac{1}{2}(\epsilon'_2 - \epsilon''_2)\right| \\ &\leq \alpha_2 + \tfrac{1}{2}\alpha_1 \ .\end{aligned}$$

The previous inequalities show that when α_1 and α_2 are negligible then so are α_3 and α_4. □

Theorem 3 (k-QR + k-SJS $\implies$ Gap 2^k-Res). *For RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$, the* Gap 2^k-Res *assumption holds if the k-*QR *assumption and the k-*SJS *assumption hold. More precisely, for any probabilistic polynomial-time distinguisher $\mathcal{B}$ against the former, there exist a k-*QR *distinguisher $\mathcal{D}_1$ and a k-*SJS *distinguisher $\mathcal{D}_2$ with comparable running times and for which*

$$\mathbf{Adv}^{\text{Gap } 2^k\text{-Res}}_{\mathcal{B}}(\kappa) \leq \tfrac{3}{2}\left((k - \tfrac{1}{3}) \cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}_1}(\kappa) + (k-1) \cdot \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}_2}(\kappa)\right) \ .$$

Proof. To prove the result, we consider a sequence of distributions which will help us bridge the gap between the assumptions. More precisely, for $0 \leq i \leq k-1$, we consider the subsets D_i of $\mathbb{J}_N$ given by

$$D_i = \{y^{2^i} \bmod N \mid y \in \mathbb{J}_N \setminus \mathbb{QR}_N\} \ .$$

We also need other subsets which can be seen as the complement of D_i in the set of 2^i-th residues that are not 2^{i+1}-th residues:

$$D'_i = \{y^{2^i} \bmod N \mid y \in \bar{\mathbb{J}}_N\} \ .$$

Finally we define the subgroup of 2^k-th residues, $R_k = \{y^{2^k} \bmod N \mid y \in \mathbb{Z}^*_N\}$.

If we consider the sets V_0 and V_1 (presented in Definition 4), we have $V_0 = D_0$ and $V_1 = R_k$. The proof will actually proceed by showing the computational indistinguishability of the (uniform) distributions induced by the corresponding subsets. Namely, unless either the k-QR* assumption or the k-SJS* assumption is false, we will prove

$$D_0 \overset{c}{\approx} D'_1 \overset{c}{\approx} D_1 \overset{c}{\approx} D'_2 \overset{c}{\approx} D_2 \overset{c}{\approx} \cdots \overset{c}{\approx} D'_{k-1} \overset{c}{\approx} D_{k-1} \ ,$$

where the $\overset{c}{\approx}$ denotes computationally indistinguishable distributions. Finally, we also prove that $D_{k-1} \overset{c}{\approx} R_k$ unless the k-QR assumption is false.

Remark 1. Note that we abuse notation by using D_i, D'_i, R_k both for subsets and for the uniform distributions over them. Also, it is important to see that:

– if $y \in_R \mathbb{J}_N \setminus \mathbb{QR}_N$ then $y^{2^i} \in_R D_i$;
– if $y \in_R \bar{\mathbb{J}}_N$ then $y^{2^i} \in_R D'_i$;
– if $y \in_R \mathbb{Z}^*_N$ then $y^{2^k} \in_R R_k$.

Claim 1. If k-QR* holds, for each $i \in \{1, \ldots, k-1\}$, no probabilistic polynomial-time adversary can distinguish the distributions of D_{i-1} and D'_i.

Proof (of Claim 1). Let $\mathcal{D}$ be a distinguisher that can tell apart D_{i-1} and D'_i with non-negligible advantage ε. We show that $\mathcal{D}$ implies a k-QR* distinguisher $\mathcal{B}_{1,i}$ with advantage ε for RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$.
Our distinguisher $\mathcal{B}_{1,i}$ takes as input an RSA modulus $N = pq$ with $p \equiv 1 \pmod{2^k}$ and an element $w \in \mathbb{Z}^*_N$ which is drawn from one of the two distributions

$$\mathsf{dist}_0 = \{y^2 \bmod N \mid y \xleftarrow{R} \mathbb{J}_N\}, \qquad \mathsf{dist}_1 = \{y \mid y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N\} .$$

Its task is to decide if w is in dist_0 or in dist_1. To this end, $\mathcal{B}_{1,i}$ chooses a random element $z \xleftarrow{R} \bar{\mathbb{J}}_N$. It then defines $x = z^{2^i} w^{2^{i-1}} \bmod N$ and feeds $\mathcal{D}$ with (x, i, N). When the distinguisher $\mathcal{D}$ halts, $\mathcal{B}_{1,i}$ outputs whatever $\mathcal{D}$ outputs.
– First assume that $w = y^2 \in \mathsf{dist}_0$, for some $y \in_R \mathbb{J}_N$. We have $x = (zy)^{2^i} \bmod N$. Further, since $z \xleftarrow{R} \bar{\mathbb{J}}_N$, we have $zy \in \bar{\mathbb{J}}_N$ and thus $x \in_R D'_i$.
– Now assume that $w \in_R \mathbb{J}_N \setminus \mathbb{QR}_N$. In this case, we clearly have $x \in_R D_{i-1}$ because $x = (z^2 w)^{2^{i-1}} \bmod N$ and $z^2 w \in \mathbb{J}_N \setminus \mathbb{QR}_N$. ∎

Claim 2. If k-SJS* holds, for each $i \in \{1, \ldots, k-1\}$, no probabilistic polynomial-time adversary can distinguish the distributions of D'_i and D_i.

Proof (of Claim 2). Let $\mathcal{D}$ be a distinguisher with non-negligible advantage ε between D_i and D'_i. We show that $\mathcal{D}$ implies a k-SJS* distinguisher $\mathcal{B}_{2,i}$ with advantage ε for RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$. Given $w \in \mathbb{Z}^*_N$ which is drawn from one of the two distributions

$$\mathsf{dist}_0 = \{y^2 \bmod N \mid y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N\}, \qquad \mathsf{dist}_1 = \{y^2 \bmod N \mid y \xleftarrow{R} \bar{\mathbb{J}}_N\},$$

$\mathcal{B}_{2,i}$ constructs $x = w^{2^{i-1}} \bmod N$ which is used to feed the distinguisher $\mathcal{D}$. When the latter outputs a result, $\mathcal{B}_{2,i}$ produces the same output. It is clear that, if $w \in_R \mathsf{dist}_0$ (resp. $w \in_R \mathsf{dist}_1$), then $x \in_R D_i$ (resp. $x \in_R D'_i$). Hence, if $\mathcal{D}$ is a successful distinguisher, so is $\mathcal{B}_{2,i}$. ∎

Claim 3. If k-QR holds, no probabilistic polynomial-time adversary can distinguish the distributions of D_{k-1} and R_k.

Proof (of Claim 3). Let $\mathcal{D}$ be an algorithm that can distinguish D_{k-1} and R_k with non-negligible advantage. We build a k-QR distinguisher $\mathcal{B}_3$ out of $\mathcal{D}$ with the same advantage.
Algorithm $\mathcal{B}_3$ takes as input $N = pq$ with $p \equiv 1 \pmod{2^k}$ as well as an element $w \in \mathbb{J}_N$ with the goal of deciding whether $w \in \mathbb{QR}_N$ or $w \in \mathbb{J}_N \setminus \mathbb{QR}_N$. To do this, $\mathcal{B}_3$ simply defines $x = w^{2^{k-1}} \bmod N$ and feeds $\mathcal{D}$ with (x, k, N). When $\mathcal{D}$ halts and outputs $b \in \{0, 1\}$, $\mathcal{B}_3$ outputs the same bit.
It is easy to see that, if $w \in_R \mathbb{QR}_N$ then $w = y^2 \bmod N$ for a random $y \in_R \mathbb{Z}^*_N$, and so $x = (y^{2^k} \bmod N) \in_R R_k$ —see Remark 1. If $w \in_R \mathbb{J}_N \setminus \mathbb{QR}_N$, we immediately have $x \in_R D_{k-1}$.
∎

To conclude the proof of the theorem, we remark that, if a probabilistic polynomial-time distinguisher $\mathcal{B}$ exists for the Gap 2^k-Res assumption (i.e., if $D_0 \overset{c}{\not\approx} R_k$), then

- either $D_{k-1} \overset{c}{\not\approx} R_k$, contradicting k-QR (Claim 3); or
- there exists $1 \leq i \leq k-1$ such that $D'_i \overset{c}{\not\approx} D_{i-1}$ or $D'_i \overset{c}{\not\approx} D_i$. The above arguments show that either situation would contradict the k-QR* assumption (Claim 1) or the k-SJS* assumption (Claim 2) —or by Lemma 1, the k-QR assumption or the k-SJS assumption.

More precisely, to get the bound given in Theorem 3, we consider $\mathcal{B}'_{2,i}$ the adversary "$\mathcal{B}$" defined in Lemma 1 when "$\mathcal{A} = \mathcal{B}_{2,i}$", and we define the distinguisher $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) as follows: it picks $(\alpha, i) \overset{R}{\leftarrow} \mathcal{P}_1$ (resp. $(\alpha, i) \overset{R}{\leftarrow} \mathcal{P}_2$), where $\mathcal{P}_1$ and $\mathcal{P}_2$ are probability distributions defined as:

$$\Pr_{(X,Y)\overset{R}{\leftarrow}\mathcal{P}_1}[(X,Y) = (\alpha, i)] = \begin{cases} \frac{2}{3k-1} & \text{if } \alpha = 1 \text{ and } i \in \{1, \ldots, k-1\} \\ \frac{1}{3k-1} & \text{if } \alpha = 2 \text{ and } i \in \{1, \ldots, k-1\} \\ \frac{2}{3k-1} & \text{if } \alpha = 3 \end{cases}$$

and

$$\Pr_{(X,Y)\overset{R}{\leftarrow}\mathcal{P}_2}[(X,Y) = (\alpha, i)] = \begin{cases} \frac{1}{3k-3} & \text{if } \alpha = 1 \text{ and } i \in \{1, \ldots, k-1\} \\ \frac{2}{3k-3} & \text{if } \alpha = 2 \text{ and } i \in \{1, \ldots, k-1\} \end{cases} .$$

Then $\mathcal{D}_1$ runs $\mathcal{B}_{1,i}$ when $\alpha = 1$, $\mathcal{B}'_{2,i}$ when $\alpha = 2$, and $\mathcal{B}_3$ when $\alpha = 3$, and outputs what this latter adversary outputs. Similarly, $\mathcal{D}_2$ runs $\mathcal{B}_{\alpha,i}$, and outputs what this latter adversary outputs.

Using Lemma 1, we have:

$$\begin{aligned} \mathbf{Adv}^{\text{Gap } 2^k\text{-Res}}_{\mathcal{B}}(\kappa) &\leq \sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-QR}^\star}_{\mathcal{B}_{1,i}}(\kappa) + \sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-SJS}^\star}_{\mathcal{B}_{2,i}}(\kappa) + \mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}_3}(\kappa) \\ &\leq \left(\sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}_{1,i}}(\kappa) + \frac{1}{2} \sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}'_{2,i}}(\kappa) + \mathbf{Adv}^{k\text{-QR}}_{\mathcal{B}_3}(\kappa) \right) + \\ &\qquad \left(\frac{1}{2} \sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{B}_{1,i}}(\kappa) + \sum_{i=1}^{k-1} \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{B}_{2,i}}(\kappa) \right) \\ &= \frac{3k-1}{2} \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}_1}(\kappa) + \frac{3k-3}{2} \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}_2}(\kappa) . \end{aligned}$$

In addition, we note that $\mathcal{D}_1$ and $\mathcal{D}_2$ have comparable running times to $\mathcal{B}$. □

We remark that the assumption $p \equiv 1 \pmod{2^k}$ is never directly used in the proof. The assumption $p \equiv 1 \pmod{2^k}$ is just needed for the correctness of our encryption scheme. The security proof actually holds for any kind of modulus N for which the QR and the SJS assumptions hold —the k-QR and the k-SJS assumptions are just the QR and the SJS assumptions for moduli $N = pq$ such that $p \equiv 1 \pmod{2^k}$.

4.3 Semantic security

It is not hard to see that the semantic security of the scheme is equivalent to the Gap 2^k-Res assumption. From Theorem 3, we thus obtain the result announced in Theorem 1. Namely, for any IND-CPA adversary $\mathcal{A}$, there exist a k-QR distinguisher $\mathcal{D}_1$ and a k-SJS distinguisher $\mathcal{D}_2$ such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{A}}(\kappa) \leq \tfrac{3}{2} \left((k - \tfrac{1}{3}) \cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}_1}(\kappa) + (k-1) \cdot \mathbf{Adv}^{k\text{-SJS}}_{\mathcal{D}_2}(\kappa) \right) .$$

Proof (of Theorem 1). The proof proceeds by simply changing the distribution of the public key. Under the Gap 2^k-Res assumption, instead of picking y uniformly in $\mathbb{J}_N \setminus \mathbb{QR}_N$, we can choose it in the subgroup of 2^k-th residues without the adversary noticing. However, in this case, the ciphertext carries no information about the message and the IND-CPA security follows. □

Interestingly, the security proof implicitly shows that, like the original Goldwasser-Micali system, our scheme is a *lossy* encryption scheme [BHY09] (i.e., it admits an alternative distribution of public keys for which encryptions statistically hide the plaintext), which provides security guarantees against selective-opening attacks [DNRS03]. Moreover, for a lossy key (y, N), there exists an efficient algorithm that opens a given ciphertext c to any arbitrary plaintext m (by using the factorization of N to find random coins that explain c as an encryption of m). It implies that our scheme satisfies the simulation-based definition [BHY09] of selective-opening security.

5 Implementation and Performance

We tackle here some implementation aspects. We explain how to select the parameters involved in the system set-up and key generation. We present fast decryption algorithms. Finally, we discuss the ciphertext expansion and give a comparison with previous schemes.

5.1 Parameter selection

The key generation (cf. Section 3.1) requires a prime p such that $p \equiv 1 \pmod{2^k}$ for some $k \geq 1$ and a random element $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$, where $N = pq$. The condition $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$ is equivalent to $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. Since a random nonzero element modulo p has a probability of exactly $\frac{1}{2}$ of being a quadratic non-residue modulo p (and similarly modulo q), a suitable y is likely to be obtained after just a few trials. Efficient algorithms for generating a prime p lying in a prescribed interval $[p_{\min}, p_{\max}]$ can be found in [JPV00, JP06]. They can be adapted to accommodate the extra condition $p \equiv 1 \pmod{2^k}$ without increasing the time complexity, as a random number congruent to 1 modulo 2^k in $[p_{\min}, p_{\max}]$ is prime with approximatively the same probability than a random odd number in $[p_{\min}, p_{\max}]$, thanks to Dirichlet's theorem. We describe such a variant below.

The goal is to produce a prime $p = 1 + 2^k r$ for some $r \in [r_{\min}, r_{\max}]$, where $r_{\min} = \lceil (p_{\min} - 1)/2^k \rceil$ and $r_{\max} = \lfloor (p_{\max} - 1)/2^k \rfloor$. Let $\Pi = 3 \cdot 5 \cdot 7 \cdots \leq r_{\max} - r_{\min} + 1$ denote a product of small odd primes. The algorithm will construct candidate primes that are automatically co-prime to Π. The first step is to generate a random unit $v \in \mathbb{Z}_\Pi^*$ (for example using the efficient algorithm presented in [JP06, § 2.2]). Define $\vartheta_0 = -\left(\frac{1}{2^k} + r_{\min}\right) \bmod \Pi$. A candidate p is then formed as

$$p \leftarrow 1 + 2^k (r_{\min} + \vartheta) \quad \text{for some } \vartheta \in_R [0, r_{\max} - r_{\min}] \text{ such that } \vartheta \equiv \vartheta_0 + v \pmod{\Pi}$$

and tested for primality. If candidate p is not prime, v is updated as $v \leftarrow 2v \bmod \Pi$ and the process is re-iterated. Since Π is odd, $2 \in \mathbb{Z}_\Pi^*$ and thus v remains in $\mathbb{Z}_\Pi^*$ after the updating step. Moreover, reducing candidate p modulo Π, we get $p \equiv 1 + 2^k (r_{\min} + \vartheta) \equiv 1 + 2^k (r_{\min} + \vartheta_0 + v) \equiv 2^k v \pmod{\Pi}$ and thus $p \in \mathbb{Z}_\Pi^*$ since $v \in \mathbb{Z}_\Pi^*$ and $2^k \in \mathbb{Z}_\Pi^*$. Equivalently, $p \in \mathbb{Z}_\Pi^*$ means that candidate p is such that $\gcd(p, p_i) = 1$ for all primes p_i dividing Π (and p is also odd by construction).

A powerful LLL-based technique due to Coppersmith bounds the size of k to at most $\frac{1}{2} \log_2 p$ bits as, otherwise, the factors of N would be revealed [Cop97, Theorem 5]. Going beyond polynomial-time

attacks, one should add an extra security margin to take into account exhaustive searches [Ngu09]. RSA moduli being balanced (i.e., $\frac{1}{2}\log_2 p = \frac{1}{4}\log_2 N$), we so end up with the upper bound

$$k < \tfrac{1}{4}\log_2 N - \kappa$$

where κ is the security parameter.

In practice, this restriction on k is not a limitation because, as described in the next section, long messages can be encrypted using the KEM/DEM paradigm. For example, using ECRYPT 2 recommendations [ECR12], for $\kappa = 128$ bits of security, a symmetric key of $k = 128$ bits has to be used for the KEM/DEM paradigm, and a 3248-bit modulus N has to be used to ensure factorization is hard. These parameters do not take into account the tightness of the reduction. If we take it into account, when $q \equiv 3 \pmod 4$, according to Theorem 2, a factor $(k+1)/2 \approx 64 = 2^6$ is lost in the reduction. Assuming that the best way to solve the quadratic residuosity consists in factorizing the modulus N, a 3584-bit modulus has to be used, as this corresponds to $(128 + 6)$ bits of security for factorization, according to [ECR12]. Note that the choice of parameters $k = 128$ and $|N|_2 = 3584$ satisfies the relation $k < \frac{1}{4}\log_2 N - \kappa$.

5.2 Optimized decryption algorithms

In its most basic version, the decryption requires $O(k)$ full modular exponentiations in $\mathbb{Z}_p^*$ in order to compute higher power residue symbols. This section shows that a suitable pre-processing phase allows increasing the decryption speed.

The RSA modulus used in the proposed cryptosystem is of the form $N = pq$ with $p \equiv 1 \pmod{2^k}$. Hence, we can write $p = 2^K p' + 1$ for some integer $K \geq k$ and some *odd* integer p'. Now, given the public key $pk = \{N, y, k\}$, consider the ciphertext $c = y^m x^{2^k} \bmod N$ of message $m = \sum_{i=0}^{k-1} m_i 2^i$ with $m_i \in \{0, 1\}$. If, for $1 \leq j \leq k$, we define $\Lambda_j = 2^{K-j}p'$ then

$$\begin{aligned} c^{\Lambda_j} &\equiv \left(y^m x^{2^k}\right)^{\Lambda_j} \equiv y^{m\Lambda_j} x^{2^{K+k-j}p'} \equiv y^{m\Lambda_j \bmod 2^K p'} \equiv y^{m\Lambda_j \bmod 2^j \Lambda_j} \\ &\equiv y^{\Lambda_j (m \bmod 2^j)} \equiv y^{\Lambda_j \left(m_{j-1}2^{j-1} + (m \bmod 2^{j-1})\right)} \equiv \left(y^{\frac{p-1}{2}}\right)^{m_{j-1}} y^{\Lambda_j (m \bmod 2^{j-1})} \\ &\equiv (-1)^{m_{j-1}} y^{\Lambda_j (m \bmod 2^{j-1})} \pmod p \ . \end{aligned}$$

So, letting $C = c^{2^{K-k}p'} \bmod p$ and $Y = y^{2^{K-k}p'} \bmod p$, the previous relation becomes $\left(\frac{C}{Y^{m \bmod 2^{j-1}}}\right)^{2^{k-j}} \equiv (-1)^{m_{j-1}} \pmod p$. Starting at $j = 1$ and iterating until $j = k$, it yields a decryption algorithm producing one bit of plaintext m per iteration (i.e., bit m_{j-1}).

To further speed-up the decryption, observing that $Y = y^{2^{K-k}p'} \bmod p$ is independent of the ciphertext, its value —or better its inverse— can be pre-computed. The private key now consists of the pair (p, D) where $D = y^{-2^{K-k}p'} \bmod p$. As one bit of plaintext m is correctly obtained per iteration, there is no need to fully recompute $D^{m \bmod 2^{j-1}} \bmod p$ at iteration j. Rather, it can be obtained more efficiently from the value of the previous iteration as

$$D^{m \bmod 2^{j-1}} \bmod p = \begin{cases} D^{m \bmod 2^{j-2}} \bmod p & \text{if } m_{j-1} = 0 \\ D^{m \bmod 2^{j-2}} D^{2^{j-1}} \bmod p & \text{if } m_{j-1} = 1 \end{cases} .$$

We thus obtain:

Algorithm 1 Decryption algorithm

Input: Ciphertext c, private key (p, D) with $D = y^{-(p-1)/2^k} \bmod p$, and public-key element k
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

1: $\mathsf{m} \leftarrow 0$; $\mathsf{B} \leftarrow 1$; $\mathsf{D} \leftarrow D$
2: $\mathsf{C} \leftarrow c^{(p-1)/2^k} \bmod p$
3: **for** $j = 1$ to $k - 1$ **do**
4: $\quad \mathsf{z} \leftarrow \mathsf{C}^{2^{k-j}} \bmod p$
5: $\quad$ **if** $(\mathsf{z} \neq 1)$ **then** $\mathsf{m} \leftarrow \mathsf{m} + \mathsf{B}$; $\mathsf{C} \leftarrow \mathsf{C} \cdot \mathsf{D} \bmod p$
6: $\quad \mathsf{B} \leftarrow 2\mathsf{B}$; $\mathsf{D} \leftarrow \mathsf{D}^2 \bmod p$
7: **end for**
8: **if** $(\mathsf{C} \neq 1)$ **then** $\mathsf{m} \leftarrow \mathsf{m} + \mathsf{B}$
9: **return** m

Variable m in the for-loop contains the lowest part of the plaintext m and variable B contains the successive powers of 2. Further, the for-loop is only performed until iteration $k - 1$ to save a couple of operations. As a variant, we remark that D can be initialized to $y^{-(p-1)/2^k} \bmod p$ (Line 1 in Alg. 1) instead of being explicitly included in the private key.

As described, the for-loop in Alg. 1 on average involves $\sum_{j=1}^{k-1}(k-j) = \frac{(k-1)k}{2}$ modular squarings for the successive evaluation of z, $\frac{k-1}{2}$ modular multiplications for the evaluation of C, and $(k-1)$ modular squarings for updating D.

Remark 2. The decryption can even be made slightly faster. The condition $\mathsf{z} \neq 1$ is equivalent to $\mathsf{z} \equiv -1 \pmod{p}$. Instead of iteratively evaluating $\mathsf{z} \leftarrow \mathsf{C}^{2^{k-j}} \bmod p$ for $1 \leq j \leq k-1$, we can set z to C and successively square it, $\mathsf{z} \leftarrow \mathsf{z}^2 \bmod p$, until it becomes congruent to $-1 \pmod{p}$. We then update C by multiplying it by the corresponding power of D and redo the process until C becomes equal to 1. On average, this halves the number of squarings for the successive evaluations of z. Furthermore, the modular squarings for updating D can be saved by pre-computing the different powers of D. This saves $(k-1)$ modular squarings. The total number of operations in the for-loop then boils down to $\frac{(k-1)k}{4}$ squarings plus $\frac{k-1}{2}$ multiplications (on average), modulo p.

5.3 Ciphertext expansion

Hybrid encryption allows designing efficient asymmetric schemes, as suggested by Shoup in the ISO 18033-2 standard for public-key encryption [ISO06]. An asymmetric cryptosystem is used to encrypt a secret key that is then used to encrypt the actual message. This is the so-called *KEM/DEM paradigm*.

The next table compares the ciphertext expansion in the encryption of k-bit messages for different generalized Goldwasser-Micali cryptosystems. Only cryptosystems with a formal security analysis are considered. Further, the value of k is assumed to be relatively small (e.g., 128 or 256) as the "message" being encrypted is typically a symmetric key (for example a 128- or 256-bit AES key) in a KEM/DEM construction.

It appears that the Goldwasser-Micali cryptosystem has the highest ciphertext expansion but its semantic security relies on the *standard* quadratic residuosity assumption (i.e., RSA moduli

Table 1. Ciphertext expansion in a typical encryption

Encryption scheme	Assumption	Ciphertext size
Goldwasser-Micali [GM84]	Quadratic residuosity (QR)	$k \cdot \log_2 N$
Benaloh-Fisher [CF85]	Prime residuosity (PR)	$\left\lceil \frac{k}{\log_2 r} \right\rceil \cdot \log_2 N$
Naccache-Stern [NS98]	Prime residuosity (PR)	$\log_2 N$
Okamoto-Uchiyama [OU98]	p-subgroup	$\log_2 N$
Paillier [Pai99]	N-th residuosity	$2\log_2 N$
This paper when $q \equiv 1 \pmod 4$	Quadratic residuosity (k-QR) + Squared Jacobi symbol (k-SJS)	$\log_2 N$
This paper when $q \equiv 3 \pmod 4$	Quadratic residuosity (k-QR)	$\log_2 N$

$N = pq$ involves *form-free* primes). The ciphertext expansion of the Benaloh-Fischer cryptosystem is similar to that of the Naccache-Stern cryptosystem for *small* messages; i.e., when $k \leq \log_2 r$. For larger messages, the Naccache-Stern cryptosystem should be preferred. It also offers the further advantage of providing a faster decryption procedure. The same is true for the Okamoto-Uchiyama cryptosystem and the Paillier cryptosystem. These two latter cryptosystems are particularly suited to encrypt very large messages (i.e., up to $\frac{1}{2}\log_2 N$ bits for the Okamoto-Uchiyama cryptosystem and up to $\log_2 N$ bits for the Paillier cryptosystem).

The encryption scheme proposed in this paper has the same ciphertext expansion as in the Naccache-Stern cryptosystem. Moreover, its decryption algorithm is fast (no searches are needed), requires less memory, and the security relies on a quadratic residuosity assumption (i.e., k-QR) when $q \equiv 3 \pmod 4$. When $q \equiv 1 \pmod 4$, it additionally requires the k-SJS assumption.

6 More Efficient Lossy Trapdoor Functions from the k-Quadratic Residuosity Assumption

In this section, we show that our homomorphic cryptosystem allows constructing a lossy trapdoor function based on the k-QR, k-SJS and DDH assumptions (or on the k-QR and DDH assumptions) with much shorter outputs and keys than in previous QR-based or DDH-based examples.

In comparison with the function of Hemenway and Ostrovsky [HO12], for example, its output is k times smaller when working with a modulus $N = pq$ with $p \equiv 1 \pmod{2^k}$. Moreover, the size of the evaluation key is decreased by a factor of $O(k^2)$ while increasing the lossiness by more than k bits. Finally, our inversion trapdoor has constant size, whereas [HO12] uses a trapdoor of size $O(n)$ to recover n-bit inputs. Our function also compares favorably with the QR-based function of Freeman *et al.* [FGK+10, FGK+13], which only loses a single bit.

In fact, by appropriately tuning our construction, we obtain the first lossy trapdoor function with short outputs, description and trapdoor that loses many input bits and relies on another assumption than Paillier's. Among known lossy trapdoor functions based on traditional number-theoretic assumptions [PW08, BFO08, FGK+10, FGK+13, KOS10, HO12, MY10], this appears as a rare efficiency tradeoff. To the best of our knowledge, it has only been achieved under the Composite Residuosity assumption [BFO08, FGK+10, FGK+13] so far.

Interestingly, our LTDF provides similar efficiency improvements to the QR-based deterministic encryption scheme of Brakerski and Segev [BS11], which also builds on the Hemenway-Ostrovsky LTDF. Note that the scheme of [BS11] is important in the deterministic encryption literature since it

is one of the only known schemes providing security in the auxiliary input setting in the standard model.

6.1 Description and security analysis

We start by recalling the following definition.

Definition 7 ([PW08]). *Let $\kappa \in \mathbb{N}$ be a security parameter and $n : \mathbb{N} \to \mathbb{N}$, $\ell : \mathbb{N} \to \mathbb{R}$ be non-negative functions of κ. A collection of (n, ℓ)-*lossy trapdoor functions *(LTDF) is a tuple of efficient algorithms* $(\mathsf{InjGen}, \mathsf{LossyGen}, \mathsf{Eval}, \mathsf{Invert})$ *with the following specifications.*

- Sampling an injective function: *Given a security parameter κ, the randomized algorithm* $\mathsf{InjGen}(1^\kappa)$ *outputs the index ek of an injective function of the family and an inversion trapdoor t.*
- Sampling a lossy function: *Given a security parameter κ, the probabilistic algorithm* $\mathsf{LossyGen}(1^\kappa)$ *outputs the index ek of a lossy function.*
- Evaluation: *Given the index of a function ek —produced by either* InjGen *or* LossyGen*— and an input $x \in \{0,1\}^n$, the evaluation algorithm* Eval *outputs $F_{ek}(x)$ such that:*
  - *If ek is an output of* InjGen*, then $F_{ek}(\cdot)$ is an injective function.*
  - *If ek was produced by* LossyGen*, then $F_{ek}(\cdot)$ has image size $2^{n-\ell}$. In this case, the value $n - \ell$ is called* residual leakage*.*
- Inversion: *For any pair (ek, t) produced by* InjGen *and any input $x \in \{0,1\}^n$, the inversion algorithm* Invert *returns $F_{ek}^{-1}(t, F_{ek}(x)) = x$.*
- Security: *The two ensembles $\{ek \mid (ek, t) \leftarrow \mathsf{InjGen}(1^\kappa)\}_{\kappa\in\mathbb{N}}$ and $\{ek \mid ek \leftarrow \mathsf{LossyGen}(1^\kappa)\}_{\kappa\in\mathbb{N}}$ are computationally indistinguishable.*

Our construction goes as follows.

Sampling an injective function. Given a security parameter κ, let $\ell_N := \ell_N(\kappa)$ and $k := k(\kappa)$ be parameters determined by κ. Let also $n := n(\kappa)$ be the desired input length. Algorithm InjGen defines $m = n/k$ (we assume that k divides n for simplicity) and conducts the following steps.

1. Generate an ℓ_N-bit RSA modulus $N = pq$ such that $p = 2^K p' + 1$ and $q = 2^L q' + 1$, for odd prime integers p, q, p', q' and with $K = k$ and $L \in \{1, \ldots, k\}$. Choose $y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$ at random.
2. For each $i \in \{1, \ldots, m\}$, pick h_i in the subgroup of 2^k-residues, $R_k = \{w^{2^k} \bmod N \mid w \in \mathbb{Z}_N^*\}$ (of order $p'q'$), by setting $h_i = {g_i}^{2^k} \bmod N$ for a randomly chosen $g_i \xleftarrow{R} \mathbb{Z}_N^*$.
3. Choose $r_1, \ldots, r_m \xleftarrow{R} \mathbb{Z}_{p'q'}$ and compute a matrix $Z = (Z_{i,j})_{i,j\in\{1,\ldots,m\}}$ given by

$$Z = \begin{pmatrix} y^{z_{1,1}} \cdot {h_1}^{r_1} \bmod N & \ldots\ldots & y^{z_{1,m}} \cdot {h_m}^{r_1} \bmod N \\ \vdots & & \vdots \\ y^{z_{m,1}} \cdot {h_1}^{r_m} \bmod N & \ldots\ldots & y^{z_{m,m}} \cdot {h_m}^{r_m} \bmod N \end{pmatrix},$$

   where $(z_{i,j})_{i,j\in\{1,\ldots,m\}}$ denotes the identity matrix.

The evaluation key is $ek := (N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}})$ and the trapdoor is $t := \{p, y\}$.

Sampling a lossy function. The process followed by LossyGen is identical to the above one but the matrix $(z_{i,j})_{i,j\in\{1,\ldots,m\}}$ is replaced by the all-zeroes $m \times m$ matrix.

Evaluation. Given $ek = (N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}})$, algorithm Eval parses the input $x \in \{0,1\}^n$ as a vector of k-bit blocks $\tilde{x} = (x_1, \ldots, x_m)$, with $x_i \in \mathbb{Z}_{2^k}$ for each i. Then, it computes and returns $\tilde{y} = (y_1, \ldots, y_m)$, with $y_j \in \mathbb{Z}_N^*$, where

$$\begin{aligned}\tilde{y} &= \Big(\prod_{i=1}^{m} Z_{i,1}{}^{x_i} \bmod N, \ldots, \prod_{i=1}^{m} Z_{i,m}{}^{x_i} \bmod N\Big) \\ &= \Big(y^{\sum_{i=1}^m z_{i,1}x_i} \cdot h_1{}^{\sum_{i=1}^m r_i x_i} \bmod N, \ldots, y^{\sum_{i=1}^m z_{i,m}x_i} \cdot h_m{}^{\sum_{i=1}^m r_i x_i} \bmod N\Big)\ .\end{aligned}$$

Inversion. Given $t = \{p, y\}$ and $\tilde{y} = (y_1, \ldots, y_m) \in \mathbb{Z}_N^m$, Invert applies the decryption algorithm of Section 3.2 to each y_j, for $j = 1$ to m. Observe that when $(z_{i,j})_{i,j\in\{1,\ldots,m\}}$ is the identity matrix, $\left(\frac{y_j}{p}\right)_{2^k} = \left[\left(\frac{y}{p}\right)_{2^k}\right]^{x_j}$ mods p. From the resulting vector of plaintexts $\tilde{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_{2^k}^m$, it recovers the input $x \in \{0,1\}^n$.

The Hemenway-Ostrovsky construction of [HO12] is slightly different in that, as in the DDH-based construction of Peikert and Waters [PW08], the evaluation key includes a vector of the form $G = (g^{r_1}, \ldots, g^{r_m})^T$, where $g \in \mathbb{QR}_N$, and the trapdoor is $t = (\log_g(h_1), \ldots, \log_g(h_m))$. In their scheme, the evaluation algorithm additionally computes $\prod_{i=1}^m (g^{r_i})^{x_i}$ while the inversion algorithm does not use the factorization of N but rather performs a coordinate-wise ElGamal decryption. Here, explicitly using the factorization of N in the inversion algorithm makes it possible to process k-bit blocks at once. In addition, it allows for a very short inversion trapdoor: the inversion algorithm only needs y and the factorization of N.

We first recall the DDH assumption before giving the security theorem for our new construction.

Definition 8 (Decision Diffie-Hellman, DDH). *Given a security parameter κ, let $\mathbb{G} = \langle g \rangle$ be a (multiplicatively written) group of order n. The* Decision Diffie-Hellman (DDH) *assumption for $\mathbb{G}$ asserts that the function $\mathbf{Adv}_{\mathcal{D}}^{\mathsf{DDH}}(\kappa)$, defined as the distance*

$$\left|\Pr[\mathcal{D}(g, g^a, g^b, g^{ab}) = 1 \mid a, b \xleftarrow{R} \mathbb{Z}_n] - \Pr[\mathcal{D}(g, g^a, g^b, g^c) = 1 \mid a, b, c \xleftarrow{R} \mathbb{Z}_n]\right|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$; the probabilities are taken over the experiment of selecting at random a generator g of $\mathbb{G}$ and choosing at random $a \in \mathbb{Z}_n$, $b \in \mathbb{Z}_n$ and $c \in \mathbb{Z}_n$.

Theorem 4. *Let $\ell(\kappa) = n(\kappa) - \log_2(p'q')$. The above construction is a $(n(\kappa), \ell(\kappa))$-LTDF if the* Gap 2^k-Res *assumption holds and if the* DDH *assumption holds in the subgroup R_k of 2^k-th residues.*

We recall that $N = pq$, with $p = 2^K p' + 1$ and $q = 2^L q' + 1$. Therefore, we have:

$$n(\kappa) - \log_2(N/2^{K+L}) < \ell(\kappa) < n(\kappa) - \log_2(N/2^{K+L}) + 1\ .$$

Proof (of Theorem 4). We first prove that lossy functions are indistinguishable from injective functions. To this end, we consider a sequence of hybrid experiments. We first define an experiment $\mathbf{Exp}_0$ which is an experiment where the key generation algorithm outputs the description of an injective function with the difference that y is chosen as a 2^k-th residue instead of being drawn as $y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$. Clearly,

under the Gap 2^k-Res assumption, $\mathbf{Exp}_0$ is computationally indistinguishable from an experiment where the adversary is given the description of an injective function. Note that although $p'q'$ is used to generate the values r_j, using the approximate value $N/2^{K+L}$ instead of $p'q'$ is statistically indistinguishable. Thus knowing the factorization of N is not necessary in these experiments, and the Gap 2^k-Res assumption can be applied.

Next, for each $i^\star \in \{1, \ldots, m\}$ we define experiment $\mathbf{Exp}_{i^\star}$ as an experiment where $y \in_R R_k$ and the key generation algorithm outputs a matrix $(Z_{i,j})_{i,j}$ which encrypts a hybrid matrix $(z_{i,j})_{i,j}$ whose first $i^\star$ columns all contain zeroes whereas the last $m - i^\star$ columns are those of the $m \times m$ identity matrix.

Claim. If the DDH assumption holds in the subgroup R_k of 2^k-th residues, for each $i^\star \in \{1, \ldots, m\}$, experiment $\mathbf{Exp}_{i^\star}$ is computationally indistinguishable from Experiment $\mathbf{Exp}_{i^\star-1}$.

Proof. The claim is proved in the same way as a similar claim about the DDH-based LTDF of Peikert and Waters [PW08]. Since y lives in the cyclic subgroup R_k of 2^k-th residues, we are free to invoke the DDH assumption in R_k. Concretely, given a DDH challenge $(g, g^a, g^b, \gamma) \in (R_k)^4$, the goal is to distinguish if $\gamma = g^{ab}$ or $\gamma \xleftarrow{R} R_k$. Let $\mathcal{B}$ be an adversary that can tell apart $\mathbf{Exp}_{i^\star}$ from $\mathbf{Exp}_{i^\star-1}$ with advantage

$$\mathbf{Adv}_{\mathcal{B}}^{\mathsf{Exp}(i^\star, i^\star-1)}(\kappa) := \Big|\Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 1 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star}] - \Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 1 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star-1}]\Big| .$$

Our distinguisher $\mathcal{D}$ is defined as follows. The public key is generated by setting $h_{i^\star} = g^a$ and $h_j = g^{\alpha_j}$, with $\alpha_j \xleftarrow{R} \mathbb{Z}_{p'q'}$ for each $j \neq i^\star$. The evaluation key is generated by setting the entry $(i^\star, i^\star)$ of the matrix as $Z_{i^\star,i^\star} = y^\beta \gamma$ for a random bit $\beta \xleftarrow{R} \{0, 1\}$, while the rest of the $i^\star$-th row is obtained by setting $Z_{i^\star,j} = (g^b)^{\alpha_j}$. The rest of rows of matrix $(Z_{i,j})_{i,j}$, different from the $i^\star$-th one, are generated by choosing the exponents faithfully; namely for each $i \neq i^\star$: $Z_{i,j} = h_j^{r_i}$ for each $j \neq i$, $Z_{j,j} = h_j^{r_j}$ for each $j < i^\star$ and $Z_{j,j} = y \cdot h_j^{r_j}$ for each $j > i^\star$, with $r_j \xleftarrow{R} \mathbb{Z}_{p'q'}$ for each $j \neq i^\star$. Element $y \in R_k$ and matrix $(Z_{i,j})_{i,j}$ are given to $\mathcal{B}$, which returns its guess β' on the running experiment. Distinguisher $\mathcal{D}$ outputs 1 if $\beta' = \beta$ and 0 otherwise.

Suppose first that $\gamma = g^{ab}$. Then it is clear that the evaluation key given to $\mathcal{B}$ is distributed as in Experiment $\mathbf{Exp}_{i^\star}$ when $\beta = 0$ and as in Experiment $\mathbf{Exp}_{i^\star-1}$ when $\beta = 1$. Hence, we have $\Pr[\mathcal{D}(g, g^a, g^b, \gamma) = 1 \mid \gamma = g^{ab}] = \Pr[\beta' = \beta \mid \gamma = g^{ab}] = \frac{1}{2}\Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 0 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star}] + \frac{1}{2}\Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 1 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star-1}] = \frac{1}{2}(1 - \Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 1 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star}]) + \frac{1}{2}\Pr[\mathcal{B}(y, (Z_{i,j})_{i,j}) = 1 \mid (y, (Z_{i,j})_{i,j}) \xleftarrow{R} \mathbf{Exp}_{i^\star-1}] = \frac{1}{2} \pm \mathbf{Adv}_{\mathcal{B}}^{\mathsf{Exp}(i^\star, i^\star-1)}(\kappa)$. If now $\gamma \xleftarrow{R} R_k$ then $\mathbf{Exp}_{i^\star}$ and $\mathbf{Exp}_{i^\star-1}$ are equally distributed. This implies that $\Pr[\mathcal{D}(g, g^a, g^b, \gamma) = 1 \mid \gamma \xleftarrow{R} R_k] = \Pr[\beta' = \beta \mid \gamma \xleftarrow{R} R_k] = 1/2$. Consequently, we get $|\Pr[\mathcal{D}(g, g^a, g^b, \gamma) = 1 \mid \gamma = g^{ab}] - \Pr[\mathcal{D}(g, g^a, g^b, \gamma) = 1 \mid \gamma \xleftarrow{R} R_k]| = \mathbf{Adv}_{\mathcal{B}}^{\mathsf{Exp}(i^\star, i^\star-1)}(\kappa)$, which should be negligible under the DDH assumption. ∎

The proof now follows by remarking that, in lossy functions, the output is entirely determined by $\sum_{i=1}^m r_i x_i \bmod p'q'$, so that the image size is smaller than $p'q'$. The residual leakage is thus at most $\log_2(p'q')$ bits. □

Combining this result with Theorem 3, the security of the new trapdoor function relies on the DDH assumption in the subgroup of 2^k-th residues and additionally either the combination of the k-QR and k-SJS assumptions (when $L > 1$) or the k-QR assumption alone (when $L = 1$).

It is worth noting that, with $N = pq$ such that $p \equiv 1 \pmod{2^k}$, a side effect of working in the subgroup R_k (of order $p'q'$) is an improved lossiness. Indeed, we lose $n - \log_2(p'q')$ bits in comparison with $n - \log_2 \phi(N)$ in [HO12]. Since $\phi(N) = 2^{K+L}p'q'$, this means we lose $K + L$ more bits than by using the construction in [HO12], where $K = k$, $1 \leq L \leq k$.

The most interesting instantiations are:

- $K = L = k$: in which case we lose $2k$ more bits than [HO12] and the construction is secure under k-QR, k-SJS, and DDH in R_k;
- $K = k$ and $L = 1$: in which case we lose only k more bits than [HO12], but the k-SJS assumption is no more required.

6.2 An all-but-one trapdoor function

Using the techniques of Peikert and Waters [PW08], it is easy to construct an equally efficient all-but-one trapdoor function providing the same amount of lossiness as our lossy trapdoor function, under the same assumptions. A difference will be that, in order to enable inversion, the resulting all-but-one function will handle $k/2$ bits (instead of k) in each chunk.

First we recall the definition of an all-but-one trapdoor function. Let $\kappa \in \mathbb{N}$ be a security parameter and $n : \mathbb{N} \to \mathbb{N}$, $\ell : \mathbb{N} \to \mathbb{R}$ be non-negative functions of κ. A collection of (n, ℓ)*-all-but-one trapdoor functions* (ABO-TDF) is a tuple of efficient algorithms (BranchGen, ABOGen, Eval, Invert) with the following specifications.

- *Sampling a branch:* Given a security parameter κ, BranchGen is a randomized algorithm that outputs a branch $b \in \{0,1\}^*$ of appropriate length.
- *Sampling a function:* ABOGen is a probabilistic algorithm that takes as input a security parameter κ and a branch $b^\star$ produced by BranchGen. It outputs the description ek of a function and a trapdoor t.
- *Evaluation:* For any branch $b^\star$ produced by BranchGen, any pair (ek, t) produced by $\mathsf{ABOGen}(1^\kappa, b^\star)$, any branch b and any input $x \in \{0,1\}^n$, the evaluation algorithm Eval outputs $F_{b,ek}(x)$ such that:
  - If $b \neq b^\star$, then $F_{b,ek}(\cdot)$ is an injective function;
  - If $b = b^\star$, then $F_{b^\star,ek}(\cdot)$ has image size $2^{n-\ell}$. In this case, the value $n - \ell$ is called *residual leakage*.
- *Inversion:* For any $b^\star$ produced by BranchGen and any pair (ek, t) produced by $\mathsf{ABOGen}(1^\kappa, b^\star)$, any branch $b \neq b^\star$ and any input $x \in \{0,1\}^n$, the inversion algorithm Invert returns $F^{-1}_{b,ek}(t, F_{b,ek}(x)) = x$.
- *Security:* For any distinct $b, b' \in \{0,1\}^*$ produced by BranchGen, the ensembles

$$\{ek \mid (ek, t) \leftarrow \mathsf{ABOGen}(1^\kappa, b)\}_{\kappa \in \mathbb{N}} \quad \text{and} \quad \{ek \mid (ek, t) \leftarrow \mathsf{ABOGen}(1^\kappa, b')\}_{\kappa \in \mathbb{N}}$$

  are computationally indistinguishable.

Our ABO-TDF is described below. A difference with the Paillier-based construction of [FGK+10] is that, when inverting the function, we must pay attention to the fact that the output of the function may contain encryptions of values which are not invertible modulo 2^k. In order to avoid the need to invert in $\mathbb{Z}_{2^k}$, we perform the division over the integers. To this end, we have to adjust the parameter k so as to make sure that, for any branches $b, b^\star$ and any input block x, the product $(b - b^\star) \cdot x$ will be smaller than 2^k.

Sampling a branch. Given a security parameter $\kappa \in \mathbb{N}$ and a parameter $\lambda := \lambda(\kappa)$ determined by κ, the algorithm chooses $b \xleftarrow{R} \{0,1\}^\lambda$.

Sampling a function. The function sampling algorithm takes as input a security parameter κ, parameters $\ell := \ell_N(\kappa)$ and $\lambda := \lambda(\kappa)$ that are determined by κ, the desired input length $n := n(\kappa)$, and a branch $b^\star \in \{0,1\}^\lambda$. It sets $k = 2\lambda$ and defines $m = n/\lambda$ (we assume that λ divides n for simplicity) and does the following.

1. Generate an ℓ_N-bit RSA modulus $N = pq$ such that $p = 2^K p' + 1$ and $q = 2^L q' + 1$, for odd prime integers p, q, p', q', $K = k$, and some $L \in \{1, \ldots, k\}$. Choose $y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$ at random.
2. For each $i \in \{1, \ldots, m\}$, pick h_i in the subgroup R_k (of order $p'q'$), by setting $h_i = {g_i}^{2^k} \bmod N$ for a randomly chosen $g_i \xleftarrow{R} \mathbb{Z}_N^*$.
3. Choose $r_1, \ldots, r_m \xleftarrow{R} \mathbb{Z}_{p'q'}$ and compute a matrix

$$Z = (Z_{i,j})_{i,j \in \{1,\ldots,m\}} = \begin{pmatrix} y^{-z_{1,1}b^\star} \cdot {h_1}^{r_1} \bmod N & \ldots\ldots & y^{z_{1,m}} \cdot {h_m}^{r_1} \bmod N \\ \vdots & & \vdots \\ y^{z_{m,1}} \cdot {h_1}^{r_m} \bmod N & \ldots\ldots & y^{-z_{m,m}b^\star} \cdot {h_m}^{r_m} \bmod N \end{pmatrix},$$

where $(z_{i,j})_{i,j \in \{1,\ldots,m\}}$ is the identity matrix; i.e., $Z_{i,i} = y^{-b^\star} {h_i}^{r_i} \bmod N$ and $Z_{i,j} = {h_j}^{r_i} \bmod N$ if $j \neq i$.

The evaluation key of the ABO function is $ek := (N, (Z_{i,j})_{i,j \in \{1,\ldots,m\}}, y)$ and the trapdoor is $t := p$.

Evaluation. In order to evaluate the function on a branch $b \in \{0,1\}^\lambda$ for the input $x \in \{0,1\}^n$ using the evaluation key $ek = (N, (Z_{i,j})_{i,j \in \{1,\ldots,m\}}, y)$, algorithm Eval parses $x \in \{0,1\}^n$ as a vector of λ-bit blocks $\tilde{x} = (x_1, \ldots, x_m)$, with $x_i \in \mathbb{Z}_{2^\lambda}$ for each i. Then, it defines the matrix

$$\begin{aligned} Z^b &= (Z^b_{i,j})_{i,j \in \{1,\ldots,m\}} \\ &= \begin{pmatrix} y^b \cdot Z_{1,1} \bmod N & Z_{1,2} & \ldots & Z_{1,m} \\ Z_{2,1} & y^b \cdot Z_{2,2} \bmod N & \ldots & Z_{2,m} \\ \vdots & & \ddots & \vdots \\ Z_{m,1} & \ldots & \ldots & y^b \cdot Z_{m,m} \bmod N \end{pmatrix}, \end{aligned}$$

i.e., $Z^b_{i,j} = Z_{i,j}$ if $i \neq j$ and $Z^b_{i,i} = y^b \cdot Z_{i,i} \bmod N$ for each $i, j \in \{1, \ldots, m\}$. Then, it computes and returns

$$\begin{aligned} \tilde{y} &= \Big(\prod_{i=1}^m (Z^b_{i,1})^{x_i} \bmod N, \ldots, \prod_{i=1}^m (Z^b_{i,m})^{x_i} \bmod N \Big) \\ &= \Big(y^{(b-b^\star)x_1} \cdot {h_1}^{\sum_{i=1}^m r_i x_i} \bmod N, \ \ldots, \ y^{(b-b^\star)x_m} \cdot {h_m}^{\sum_{i=1}^m r_i x_i} \bmod N \Big) \ . \end{aligned}$$

INVERSION. Given a description $ek = (N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}, y)$ of the function, the trapdoor $t = p$ and the output $\tilde{y} = (y_1, \ldots, y_m) \in \mathbb{Z}_N^m$, the function can be inverted for the branch $b \neq b^\star$ by proceeding as follows.

1. Define the vector $(w_1, \ldots, w_m) \in \mathbb{Z}_N^m$ as $(w_1, \ldots, w_m) = (y_1, \ldots, y_m)$ if $b > b^\star$ (when the bitstrings b and $b^\star$ are interpreted as natural integers) and $(w_1, \ldots, w_m) = ({y_1}^{-1} \bmod N, \ldots, {y_m}^{-1} \bmod N)$ if $b < b^\star$.
2. For $i = 1$ to m, apply the decryption algorithm of Section 3.2 to w_i.
3. From the vector of plaintexts $\tilde{x} = (x_1, \ldots, x_m) \in \mathbb{Z}_{2^\lambda}^m$ obtained at Step 2, define $\tilde{x}' = (x'_1, \ldots, x'_m) \in \mathbb{Z}_{2^\lambda}^m$ such that $x'_i = x_i/|b - b^\star|$ (the division being performed over $\mathbb{Z}$), where $|b - b^\star| = b - b^\star$ if $b > b^\star$ and $b^\star - b$ otherwise.
4. From $\tilde{x}' = (x'_1, \ldots, x'_m)$, recover the original input $x \in \{0,1\}^n$ by concatenating the binary representations the coordinates of $\tilde{x}'$.

The correctness of the inversion algorithm stems from the fact that, since we have $x_i, b, b^\star < 2^\lambda$, it holds that $|b - b^\star| \cdot x_i < 2^{2\lambda} = 2^k$ for each $i \in \{1, \ldots, m\}$, so that x'_i can be computed over the integers at step 3 of the inversion algorithm.

It is easy to prove that the description of the function computationally hides the underlying lossy branch if the k-QR and k-SJS assumptions hold (when $L > 1$) or if the k-QR assumption holds (when $L = 1$), and if the DDH assumption holds in the subgroup R_k (of order $p'q'$). The proof is essentially identical to the proof of Theorem 4 and is omitted.

6.3 Application: Efficient CCA-secure encryption

By combining the lossy and all-but-one trapdoor function, a CCA-secure encryption scheme can be obtained using the construction of [PW08]. We argue now that $m = O(1)$ suffices for this purpose. Recall that the scheme of [PW08] combines a pairwise independent hash function $H : \{0,1\}^n \to \{0,1\}^\tau$, an (n, ℓ)-lossy function and an (n, ℓ')-all-but-one function such that $\ell + \ell' \geq n + \nu$ and $\tau \geq \nu - 2\log_2(1/\varepsilon)$, for some $\nu \in \omega(\log n)$ and where ε is the statistical distance in the modified Leftover Hash Lemma used in [DRS04]. If we choose $\varepsilon \approx 2^{-\kappa}$ and $\tau = k$ in order to encrypt k-bit messages, we can set $\nu = k + 2\kappa$. Setting $\ell = \ell' = n - \log_2(p'q')$, the constraint $\ell + \ell' \geq n + \nu$ translates into $n - 2\log_2(p'q') \geq \nu$.

Since $q = 2^L q' + 1$ and $p = 2^k p' + 1$ in our trapdoor functions, if we set $k = \frac{1}{4}\log_2 N - \kappa$ (cf. Section 5.1), we have $\log_2(p'q') = \log_2 \phi(N) - k - L \approx 4(k + \kappa) - k - L = 3k + 4\kappa - L$, which yields $n \geq 4k + 6\kappa - L$. If $k > \kappa$, it is sufficient to set $n \geq 10k$. If we take into account the fact that our all-but-one function processes blocks of $k/2$ bits, we find that $m = 2n/k = 20$ suffices here, even for $L = 1$. For larger values of L, an even smaller m would suffice.

As it turns out, when the Peikert-Waters construction [PW08, Section 4.3] of CCA-secure encryption is instantiated with our lossy and all-but-one trapdoor functions, it only requires a constant number of exponentiations while retaining constant-size public keys and ciphertexts.

With the exception of [HKS13] (which relies on a weaker assumption), to the best of our knowledge, it yields the only known CCA-secure QR-based cryptosystem combining the aforementioned efficiency properties. Up to now, the most efficient chosen-ciphertext-secure cryptosystem strictly based on the QR assumption was the one of Kiltz *et al.* [KPSY09], where $O(\kappa)$ exponentiations are needed to encrypt and the public key contains $O(\kappa)$ group elements. On the other hand, our construction requires more specific moduli than [KPSY09] and additionally appeals to the DDH assumption (and the k-SJS assumption, as well, if $L > 1$).

7 Conclusion

This paper introduced a new generalization of the Goldwasser-Micali cryptosystem. The so-obtained cryptosystems are shown to be secure under well-defined assumptions. Further, they enjoy a number of useful features including fast decryption, optimal ciphertext expansion, and homomorphic property. We believe that our proposal is the most natural yet efficient generalization of the Goldwasser-Micali cryptosystem. It keeps the nice attributes and properties of the original scheme while improving the overall performance.

When applied to the Peikert-Waters framework for building lossy trapdoor functions, it yields a practical construction based on quadratic-residuosity related and DDH assumptions, with companion deterministic encryption scheme and CCA-secure cryptosystem.

Acknowledgments

The authors are thankful to an anonymous referee for useful comments. The third author is also thankful to Antoine Joux for comments on an earlier version of this work.

The work of the second author is partially supported by project MTM 2013-41426-R of Spanish Ministry MINECO.

References

ABP13. M. Abdalla, F. Ben Hamouda, and D. Pointcheval. Tighter reductions for forward-secure signature schemes. In *PKC 2013*, *LNCS* 7778, pages 292–311. Springer, February / March 2013.

BBN+09. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT 2009*, *LNCS* 5912, pages 232–249. Springer, December 2009.

BBO07. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO 2007*, *LNCS* 4622, pages 535–552. Springer, August 2007.

BBS82. L. Blum, M. Blum, and M. Shub. Comparison of two pseudo-random number generators. In *CRYPTO'82*, pages 61–78. Plenum Press, New York, USA, 1982.

BBS86. L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM J. Comput.*, 15(2):363–383, 1986.

Ben87. J. D. C. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, New Haven, CT, USA, 1987.

BFO08. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *CRYPTO 2008*, *LNCS* 5157, pages 335–359. Springer, August 2008.

BG84. M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *CRYPTO'84*, *LNCS* 196, pages 289–302. Springer, August 1984.

BHY09. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT 2009*, *LNCS* 5479, pages 1–35. Springer, April 2009.

BS11. Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *CRYPTO 2011*, *LNCS* 6841, pages 543–560. Springer, August 2011.

CF85. J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *26th FOCS*, pages 372–382. IEEE Computer Society Press, October 1985.

CGHGN01. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen. Paillier's cryptosystem revisited. In *ACM CCS 01*, pages 206–214. ACM Press, November 2001.

Cop97. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

DJN10. I. Damgård, M. Jurik, and J. B. Nielsen. A generalization of Paillier's public-key system with applications to electronic voting. *Int. J. Inf. Sec.*, 9(6):371–385, 2010.

DNRS03. C. Dwork, M. Naor, O. Reingold, and L. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

DRS04. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT 2004*, *LNCS* 3027, pages 523–540. Springer, May 2004.

ECR12. ECRYPT II. Yearly report on algorithms and keysizes, 2012.

FGK+10. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In *PKC 2010*, *LNCS* 6056, pages 279–295. Springer, May 2010.

FGK+13. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 26(1):39–74, January 2013.

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

Gol04. O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2004.

Gro05. J. Groth. Cryptography in subgroups of $\mathbb{Z}_n$. In *TCC 2005*, *LNCS* 3378, pages 50–65. Springer, February 2005.

HKS13. D. Hofheinz, E. Kiltz, and V. Shoup. Practical chosen ciphertext secure encryption from factoring. *J. Cryptology*, 26(1):102–118, January 2013.

HO12. B. Hemenway and R. Ostrovsky. Extended-DDH and lossy trapdoor functions. In *PKC 2012*, *LNCS* 7293, pages 627–643. Springer, May 2012.

IR90. K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory*, *Graduate Texts in Mathematics* 84. Springer, 2nd edition, 1990.

ISO06. ISO/IEC 18033-2. Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers. International Organization for Standardization, May 2006.

JP06. M. Joye and P. Paillier. Fast generation of prime numbers on portable devices: An update. In *CHES 2006*, *LNCS* 4249, pages 160–173. Springer, October 2006.

JPV00. M. Joye, P. Paillier, and S. Vaudenay. Efficient generation of prime numbers. In *CHES 2000*, *LNCS* 1965, pages 340–354. Springer, August 2000.

KKOT90. K. Kurosawa, Y. Katayama, W. Ogata, and S. Tsujii. General public key residue cryptosystems and mental poker protocols. In *EUROCRYPT'90*, *LNCS* 473, pages 374–388. Springer, May 1990.

KL07. J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. CRC Press, 2007.

KOS10. E. Kiltz, A. O'Neill, and A. Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In *CRYPTO 2010*, *LNCS* 6223, pages 295–313. Springer, August 2010.

KPSY09. E. Kiltz, K. Pietrzak, M. Stam, and M. Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT 2009*, *LNCS* 5479, pages 590–609. Springer, April 2009.

Lem00. F. Lemmermeyer. *Reciprocity Laws*. Springer Monographs in Mathematics. Springer, 2000.

MV04a. J. Monnerat and S. Vaudenay. Generic homomorphic undeniable signatures. In *ASIACRYPT 2004*, *LNCS* 3329, pages 354–371. Springer, December 2004.

MV04b. J. Monnerat and S. Vaudenay. Undeniable signatures based on characters: How to sign with one bit. In *PKC 2004*, *LNCS* 2947, pages 69–85. Springer, March 2004.

MY10. P. Mol and S. Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In *PKC 2010*, *LNCS* 6056, pages 296–311. Springer, May 2010.

Ngu09. P. Q. Nguyen. Public-key cryptanalysis. In *Recent Trends in Cryptography*, Contemporary Mathematics. AMS–RSME, 2009.

NS98. D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. In *ACM CCS 98*, pages 59–66. ACM Press, November 1998.

OP01. T. Okamoto and D. Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In *PKC 2001*, *LNCS* 1992, pages 104–118. Springer, February 2001.

OU98. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In *EUROCRYPT'98*, *LNCS* 1403, pages 308–318. Springer, May / June 1998.

Pai99. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, *LNCS* 1592, pages 223–238. Springer, May 1999.

PH78. S. H. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Tran. Inf. Theory*, 24(1):106–110, 1978.

PLW95. S. J. Park, B. Y. Lee, and D. H. Won. A probabilistic encryption using very high residuosity and its applications. In *Global Telecommunications Conference (GLOBECOM'95)*, pages 1179–1182. IEEE Press, 1995.

PW08. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

Reg09. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. Earlier version in STOC 2005.

Sch98. R. Scheidler. A public-key cryptosystem using purely cubic fields. *J. Cryptology*, 11(2):109–124, 1998.

Sho10. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2nd edition, 2010.
SW95. R. Scheidler and H. C. Williams. A public-key cryptosystem utilizing cyclotomic fields. *Des. Codes Cryptography*, 6(2):117–131, 1995.
Wee12. H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *EUROCRYPT 2012*, *LNCS* 7237, pages 246–262. Springer, April 2012.
Yan02. S. Y. Yan. *Number Theory for Computing*. Springer, 2nd edition, 2002.
ZMI88. Y. Zheng, T. Matsumoto, and H. Imai. Residuosity problem and its applications to cryptography. *Trans. IEICE*, E-71(8):759–767, 1988.

A Proof of Theorem 2

As in the proof of Theorem 3, for $0 \leq i \leq k-1$, we consider the subsets D_i of $\mathbb{J}_N$ given by

$$D_i = \{y^{2^i} \bmod N \mid y \in \mathbb{J}_N \setminus \mathbb{QR}_N\}$$

and define the subgroup of 2^k-th residues, $R_k = \{y^{2^k} \bmod N \mid y \in \mathbb{Z}_N^*\}$.

We start with a lemma that is useful to tighten the bound of Theorem 3 in the case $q \equiv 3 \pmod 4$.

Lemma 2. *Let $N = pq$ be the product of two large primes p and q where $p \equiv 1 \pmod{2^k}$ for some $k \geq 1$ and $q \equiv 3 \pmod 4$. Then, for any $w \in \mathbb{QR}_N$, letting $W := w^{2^{i-1}} \bmod N$ for a given $1 \leq i \leq k$, we have $W \in R_k \cup \bigcup_{j=i}^{k-1} D_j$. Further, if w is uniform over $\mathbb{QR}_N$, we have W uniform over D_j with probability $\frac{1}{2^{j-i+1}}$ for $i \leq j \leq k-1$ and W uniform over R_k with probability $\frac{1}{2^{k-i}}$.*

Proof. We assume that w is uniform over $\mathbb{QR}_N$.

The case $i = k$ (which includes the case $k = 1$) yields $W = w^{2^{k-1}}$ with $w \in \mathbb{QR}_N$. It is then readily verified that W is uniform over R_k with probability 1.

We henceforth suppose $i \leq k-1$ and $k \geq 2$. In particular, this implies $p \equiv 1 \pmod 4$ and thus $\left(\frac{-1}{p}\right) = 1$. Denoting by $(\hat{w}_p, \hat{w}_q)$ the CRT representation of a square root $\hat{w}$ of w (i.e., $\hat{w}_p = \hat{w} \bmod p$ and $\hat{w}_q = w \bmod q$), the four square roots of w modulo N are given by $(\pm\hat{w}_p, \pm\hat{w}_q)$. Since $\left(\frac{-1}{q}\right) = -1$, we can assume w.l.o.g. that $\left(\frac{\hat{w}_q}{q}\right) = \left(\frac{\hat{w}_p}{p}\right)$, or equivalently that $\hat{w} \in \mathbb{J}_N$. If $\hat{w} \in \mathbb{QR}_N$ the process can be re-iterated, and so on. More generally, we define t as the largest integer in $\{1, \ldots, k-i\}$ such that $w = \hat{w}^{2^t}$ for some $\hat{w} \in \mathbb{J}_N$. We can so write $W = \hat{w}^{2^{t+i-1}}$ for some $\hat{w} \in \mathbb{J}_N$. It is worth noting that since t is the *largest* integer in the set $\{1, \ldots, k-i\}$ we can only have $\hat{w} \in \mathbb{QR}_N$ when $t = k - i$. Defining $j = t + i - 1$ (observe that $i \leq j \leq k-1$), we therefore obtain $W = \hat{w}^{2^j} \in D_j$ if $\hat{w} \notin \mathbb{QR}_N$ (i.e., $\hat{w} \in \mathbb{J}_N \setminus \mathbb{QR}_N$) and $W = \hat{w}^{2^{k-1}} \in R_k$ if $\hat{w} \in \mathbb{QR}_N$. The probability that $W \in D_j$ (for $i \leq j \leq k-1$) is $\Pr[w = \hat{w}^{2^t} \text{ and } \hat{w} \notin \mathbb{QR}_N] = \frac{1}{2^t} = \frac{1}{2^{j-i+1}}$ and the probability that $W \in R_k$ is $\Pr[W \notin \bigcup_{j=i}^{k-1} D_j] = 1 - \sum_{j=i}^{k-1} \frac{1}{2^{j-i+1}} = \frac{1}{2^{k-i}}$. □

Theorem 5. *For RSA moduli $N = pq$ with $p \equiv 1 \pmod{2^k}$ and $q \equiv 3 \pmod 4$, the Gap 2^k-Residuosity* Gap 2^k-Res *assumption holds if the k-*QR *assumption holds. More precisely, for any probabilistic polynomial-time distinguisher $\mathcal{B}$ against the latter, there exists a k-*QR *distinguisher $\mathcal{D}$ with comparable running time and for which*

$$\mathbf{Adv}^{\text{Gap } 2^k\text{-Res}}_{\mathcal{B}}(\kappa) \leq \tfrac{1}{2}(k+1) \cdot \mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}}(\kappa) \ .$$

Proof. Let $\mathcal{B}$ be an adversary against Gap 2^k-Res. We write:

$$\epsilon_i = \begin{cases} \Pr[\mathcal{B}(x,N)=1 \mid x \xleftarrow{R} D_i] & \text{for } i \in \{0,\dots,k-1\} \\ \Pr[\mathcal{B}(x,N)=1 \mid x \xleftarrow{R} R_k] & \text{for } i = k \end{cases} .$$

The advantage of $\mathcal{B}$ against Gap 2^k-Res is given by $\mathbf{Adv}_{\mathcal{B}}^{\text{Gap } 2^k\text{-Res}}(\kappa) = |\epsilon_0 - \epsilon_k|$.

We first construct k distinguishers $\mathcal{B}_1,\dots,\mathcal{B}_k$ against k-QR as follows. $\mathcal{B}_i$ takes as input an RSA modulus $N = pq$, with $p \equiv 1 \pmod{2^k}$ and $q \equiv 3 \pmod 4$, and an element $w \in \mathbb{J}_N$. Its task is to decide whether w is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$ or uniform over $\mathbb{QR}_N$. To this end, $\mathcal{B}_i$ chooses a random element $z \xleftarrow{R} \mathbb{Z}_N^*$, defines $x = z^{2^i} w^{2^{2^i-1}} \bmod N$, and feeds $\mathcal{B}$ with (x,N). It outputs the answer returned by $\mathcal{B}$. There are two cases:

- If w is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$, we clearly have that x is uniform over D_{i-1}. Therefore, in that case, $\mathcal{B}_i$ outputs 1 with probability ϵ_{i-1}.
- If w is uniform over $\mathbb{QR}_N$, $\mathcal{B}$ outputs 1 with probability $\sum_{j=i}^{k} \frac{1}{2^{j-i+1}}\epsilon_j + \frac{1}{2^{k-i}}\epsilon_k$, according to Lemma 2.

Therefore, the (signed) advantage of $\mathcal{B}_i$ in solving k-QR is

$$a_i = \epsilon_{i-1} - \left(\sum_{j=i}^{k} \frac{1}{2^{j-i+1}}\epsilon_j + \frac{1}{2^{k-i}}\epsilon_k \right) = \epsilon_{i-1} - \sum_{j=i}^{k} 2^{i-1}\beta_j\epsilon_j$$

with $\beta_j = \frac{1}{2^j}$ for $j \in \{1,\dots,k-1\}$ and $\beta_k = \frac{1}{2^{k-1}}$.

Consider the following probability distribution $\mathcal{P}$ over $\{1,\dots,k\}$:

$$\Pr_{X \xleftarrow{R} \mathcal{P}}[X = i] = p_i := \begin{cases} \frac{2}{k+1} & \text{if } i = 1 \\ \frac{1}{k+1} & \text{if } i \geq 2 \end{cases} .$$

We now define an adversary $\mathcal{D}$ against k-QR as follows: $\mathcal{D}$ chooses a random element $i \xleftarrow{R} \mathcal{P}$ and feeds $\mathcal{B}_i$ with its k-QR challenge. The advantage of $\mathcal{D}$ is equal to

$$\begin{aligned} \mathbf{Adv}_{\mathcal{D}}^{k\text{-QR}}(\kappa) &= \left| \sum_{i=1}^{k} p_i a_i \right| = \left| \sum_{i=1}^{k} p_i\epsilon_{i-1} - \sum_{i=1}^{k} p_i \sum_{j=i}^{k} 2^{i-1}\beta_j\epsilon_j \right| = \left| \sum_{j=0}^{k-1} p_{j+1}\epsilon_j - \sum_{j=1}^{k}\sum_{i=1}^{j} 2^{i-1}p_i\beta_j\epsilon_j \right| \\ &= \left| p_1\epsilon_0 + \sum_{j=1}^{k-1}\left(p_{j+1} - \sum_{i=1}^{j} 2^{i-1}p_i\beta_j\right)\epsilon_j - \sum_{i=1}^{k} 2^{i-1}p_i\beta_k\epsilon_k \right| . \end{aligned}$$

For $j \in \{1,\dots,k-1\}$, we have $\beta_j = \frac{1}{2^j}$ and thus

$$\sum_{i=1}^{j} 2^{i-1}p_i\beta_j = \left(p_1 + \sum_{i=2}^{j} 2^{i-1}p_i\right)\beta_j = \left(2 + \sum_{i=2}^{j} 2^{i-1}\right)\frac{\beta_j}{k+1} = 2^j\,\frac{\beta_j}{k+1} = \frac{1}{k+1} = p_{j+1} .$$

Likewise, since $\beta_k = \frac{1}{2^{k-1}}$, we have

$$\sum_{i=1}^{k} 2^{i-1} p_i \beta_k = \left(p_1 + \sum_{i=2}^{k} 2^{i-1} p_i \right) \beta_k = 2^k \frac{\beta_k}{k+1} = \frac{2}{k+1} \ .$$

Therefore, we get that the advantage of $\mathcal{D}$ satisfies

$$\mathbf{Adv}^{k\text{-QR}}_{\mathcal{D}}(\kappa) = \left| \frac{2}{k+1} \epsilon_0 + 0 - \frac{2}{k+1} \epsilon_k \right| = \frac{2}{k+1} \mathbf{Adv}^{\text{Gap } 2^k\text{-Res}}_{\mathcal{B}}(\kappa) \ .$$

This concludes the proof by further noting that the running time of $\mathcal{D}$ is comparable to that of $\mathcal{B}$. □

Theorem 2 is now an application of Theorem 5, in a way similar to what was done in Section 4.3 for the general case.