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

Marc Joye[1] and Benoît Libert[2,**]

[1] Technicolor
175 South San Antonio Road, Los Altos, CA 94022, USA
marc.joye@technicolor.com
[2] 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 by Goldwasser and Micali in their seminal paper [21]. They also introduced the equivalent notion of *indistinguishability of encryptions*, which is usually easier to work with. Given the encryption 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 public-key encryption schemes. One such cryptosystem was also presented in [21]. 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 $m \in \{0, 1\}$

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

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 string $m = (m_{k-1}, \dots, m_0)_2$, 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 [8]. They achieve a better ciphertext expansion: the ciphertext has the same length as the plaintext plus an integer of the size of modulus. The scheme is proved semantically secure assuming the unpredictability of the output of the Blum-Blum-Shub's pseudorandom generator [6,7] which resides on the factorisation hardness assumption. Details about this scheme can be found in [20].

Another direction, put forward by Benaloh and Fischer [12,5], 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 [41]. 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 [22].

Other generalizations and extensions of the Goldwasser-Micali cryptosystem but without formal security analysis can be found in [55,34,46] and, more recently and concurrently to this paper, in [24] that presents essentially the same schemes but with an incomplete security analysis. In [40,39], Monnerat and Vaudenay developed applications using the more general theory of characters, specifically with characters of order ≤ 4. Related cryptosystems are described in [51,50]. Yet another, different approach was proposed by Okamoto and Uchiyama [44], 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 [45] to the setting $N = p^2 q^2$. In 2005, Boneh, Goh and Nissim [10] showed an additively homomorphic system also supporting one multiplication.

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 [47], are function families wherein injective functions are computationally indistinguishable 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 [47], deterministic encryption [2,9] as well as cryptosystems that retain some security in the absence of reliable randomness [3] or in the presence of selective-opening adversaries [4].

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 [12] and Naccache and Stern [41]. Before discussing it, we quote from [41]:

> *"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 very fast, even faster than in the Naccache-Stern cryptosystem. Searches are no longer needed (not even in smaller subspaces) in the decryption algorithm as plaintext messages can be recovered bit by bit. In its basic version, our decryption algorithm is asymptotically slower than in Paillier's cryptosystem as the number of bit operations is quartic. However, we suggest several optimizations to speed up the decryption process. In one of these, the decryption cost is dominated by $O(k)$ modular squarings and multiplications —which only takes $O(k^3)$ operations— at the expense of storing the equivalent of $O(k)$ RSA moduli throughout intermediate steps. As a last advantage, the underlying complexity assumption is 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 $p, q \equiv 1 \pmod{2^k}$ and the hardness of determining the Jacobi symbol of an element $y \in \mathbb{Z}_N^*$ given (x, N) where $x = y^2 \bmod N$.

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. An interesting extension would be to thresholdize it as was done in [31].

As another useful property, the new scheme also inherits the selective opening security[3] [16,4] of the Goldwasser-Micali system (in the sense of a simulation-based definition given in [4]). 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.

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 [47] were based on the Decision Diffie-Hellman and Learning-with-Error [49] assumptions. More efficient examples based on the Composite Residuosity assumption were given in [9,17,18] while Kiltz *et al.* [32] showed that the RSA permutation provides a lossy function. Under the quadratic residuosity assumption, three distinct constructions were put forth in [23,17,18,53]. Those of Freeman *et al.* [17,18] and of Wee [53] must be used in combination with the results of Mol and Yilek [38] as they only lose single bits of information about the input. Hemenway and Ostrovsky [23] suggested a more efficient realization, of which Wee's framework [53] is a generalization. While their QR-based LTDF has found applications in the design of deterministic encryption schemes [11], 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 the length of the output (resp. the description of the function) by a factor of $O(\kappa)$ (resp. $O(\kappa^2)$), where κ is the security parameter. By appropriately selecting the parameters, we obtain evaluation keys and outputs consisting of a constant number of $\mathbb{Z}_N^*$ elements (and thus $O(\kappa)$ bits, instead of $O(\kappa^2)$ or $O(\kappa^3)$ as in the previous constructions). We thus obtain a DDH/QR-based LTDF, whose efficiency is competitive with Paillier-based realizations [9,17,18]. 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 [11].

[3] 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 remaining ciphertexts.

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 and prove its security. Section 4 discusses certain implementation aspects. In Section 5, we describe our new lossy trapdoor function. Finally, we conclude in Section 6. Optimized decryption algorithms are presented in Appendix B.

Corrigendum As stated in the proceedings version ([28]), Theorem 3 is incomplete for the construction of LTDFs. It additionally requires the DDH assumption. This is corrected in this full version. We also correct in this version the statement of Theorem 1; the SJS assumption (see Definition 3) is missing in [28]. Finally, we note that the faster decryption algorithms of Appendix B are not present in the proceedings version. This is a new contribution.

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 [26,52,54] for further details on (quadratic) residuosity. More information about encryption schemes can be found in textbooks in cryptography; e.g. [20,30].

2.1 n^{th}-power residues

Let $N \in \mathbb{N}$. For each integer $n \geq 2$, we define $(\mathbb{Z}_N^*)^n = \{x^n \mid x \in \mathbb{Z}_N^*\}$ 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 [35]). In this paper, we consider the n-th power residue symbol for a divisor n of $(p-1)$, as presented in [54, 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*

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

is called the n-th power residue symbol modulo p, where $a^{\frac{p-1}{n}}$ mods p represents the absolute smallest residue of $a^{\frac{p-1}{n}}$ modulo p (namely, the complete set of absolute smallest residues are: $-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2$).

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

2.2 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, $\mathbb{J}_N = \left\{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{N}\right) = 1\right\}$. A relevant subset of $\mathbb{J}_N$ is the set of quadratic residues modulo N, $\mathbb{QR}_N = \left\{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1\right\}$. The set of integers whose Jacobi symbol is -1 is denoted by $\overline{\mathbb{J}}_N$; i.e., $\overline{\mathbb{J}}_N = \left\{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{N}\right) = -1\right\} = \mathbb{Z}_N^* \setminus \mathbb{J}_N$.

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 moduli $N = pq$ with $p, q \equiv 1 \pmod{2^k}$, we will refer to it as the k-QR *assumption*. Formally, we have:

Definition 2 (Quadratic Residuosity Assumption). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter* κ*, outputs primes* p *and* q *such that* $p, q \equiv 1 \pmod{2^k}$*, and their product* $N = pq$*. The* Quadratic Residuosity (k-QR) *assumption asserts that the function* $\mathbf{Adv}_{\mathcal{D}}^{k\text{-}\mathsf{QR}}(1^\kappa)$*, 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 for RSA moduli $N = pq$ when $p, q \equiv 1 \pmod 4$. Since -1 is a square modulo p and q, the square roots of any element of $\mathbb{QR}_N$ all have the same Jacobi symbol modulo N. 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 $x = y^2 \bmod N$. Formally, we define:

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

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 primes p and q are chosen congruent to 1 modulo 2^k where k denotes the bit-size of the messages being encrypted.

In more detail, our encryption scheme is the tuple (KeyGen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$) defined as follows.

$\mathsf{KeyGen}(1^\kappa)$ Given a security parameter κ, KeyGen defines an integer $k \geq 1$, randomly generates primes $p, q \equiv 1 \pmod{2^k}$, and sets $N = pq$. It also picks $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$. 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

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

holds. An efficient method to recover message m in a bit-by-bit fashion is detailed in the next section (§ 3.2). More efficient variants are provided in Appendix B.

The correctness 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 $\alpha^m \equiv z \pmod{p}$.

Remark 1. We notice that the case $k = 1$ corresponds to the Goldwasser-Micali cryptosystem. Indeed, the 2^k-th power residue symbol is then the classical Legendre symbol and the assumption $p, q \equiv 1 \pmod{2^k}$ is trivially verified.

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 [48] and is detailed below.

Algorithm 1 Decryption algorithm

Input: Ciphertext c, private key p (and public-key elements y and k)
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

```
m ← 0; B ← 1
for i = 1 to k do
    z ← (c/p)_{2^i}; t ← (y/p)_{2^i}^m mods p
    if (t ≠ z) then m ← m + B
    B ← 2B
end for
return m
```

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

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

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 $1 \le i \le k$. As a result, m can be recovered bit by bit using p, starting from the rightmost bit. The algorithm uses an accumulator B which contains the successive powers of 2.

3.3 Security analysis

The case $k = 1$ corresponds to the Goldwasser-Micali cryptosystem which has indistinguishable encryptions under the standard Quadratic Residuosity assumption. So, we henceforth assume $k \geq 2$. We will prove that for $k \geq 2$ the scheme provides indistinguishable encryptions under the k-QR and k-SJS assumptions.

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 [43]) 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). *Let* RSAGen *be a probabilistic algorithm which, given a security parameter κ, outputs primes p and q such that $p, q \equiv 1 \pmod{2^k}$, and their product $N = pq$. The* Gap 2^k-Residuosity (Gap-2^k-Res) *problem in $\mathbb{Z}_N^*$ is to distinguish the distribution of the following two sets given only $N = pq$:*

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

The Gap 2^k-Residuosity *assumption posits that the advantage $\mathbf{Adv}_{\mathcal{D}}^{\mathsf{Gap}\text{-}2^k\text{-}\mathsf{Res}}(1^\kappa)$ of any PPT distinguisher $\mathcal{D}$, defined as the distance*

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

where probabilities are taken over all coin tosses, is negligible.

The latter assumption was independently considered by Abdalla, Ben Hamouda and Pointcheval [1] who used it to provide tighter security proofs for forward-secure signatures.

In the above definition, we explicitly give k to the distinguisher and remark that this information should be of little help considering that it can always be guessed with non-negligible probability. Also observe that from $p, q \equiv 1 \pmod{2^k}$, it follows that $2^k \mid N - 1$.

We now investigate the relationship between the Gap 2^k-Residuosity assumption and other more natural assumptions; namely, we will show that it is implied by the k-QR and k-SJS assumptions. To this end, it is useful to introduce two intermediate assumptions: the "special" QR assumption and the "special" SJS assumption.

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

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

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

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

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 $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$ and $y \in \overline{\mathbb{J}}_N$.

Lemma 1. *Using the previous notation, we have* k-QR + k-$\mathsf{SJS} \implies k$-$\mathsf{QR}^\star$ + k-$\mathsf{SJS}^\star$. *More precisely, any PPT distinguisher* $\mathcal{A}$ *against* k-QR *or* k-SJS *is also a distinguisher against* k-$\mathsf{QR}^\star$ *or* k-$\mathsf{SJS}^\star$ *with advantage satisfying*

$$\mathbf{Adv}_{\mathcal{A}}^{k\text{-}\mathsf{QR}^\star+k\text{-}\mathsf{SJS}^\star}(1^\kappa) \leq 2 \cdot \mathbf{Adv}_{\mathcal{A}}^{k\text{-}\mathsf{QR}+k\text{-}\mathsf{SJS}}(1^\kappa) \ .$$

Proof. Consider a PPT algorithm $\mathcal{A}$ taking on input N and $x \in \mathbb{J}_N$. For $x \xleftarrow{R} \mathbb{J}_N$, we let

$$\epsilon_1 = \Pr[\mathcal{A}(x,N) = 1 \mid x \in \mathbb{J}_N \setminus \mathbb{QR}_N]\,, \qquad \epsilon_2' = \Pr[\mathcal{A}(x,N) = 1 \mid x = y^2 \in \mathbb{QR}_N \text{ and } y \in \mathbb{J}_N \setminus \mathbb{QR}_N]\,,$$
$$\epsilon_2'' = \Pr[\mathcal{A}(x,N) = 1 \mid x = y^2 \in \mathbb{QR}_N \text{ and } y \in \mathbb{QR}_N]\,, \quad \epsilon_3 = \Pr[\mathcal{A}(x,N) = 1 \mid x = y^2 \in \mathbb{QR}_N \text{ and } y \notin \mathbb{J}_N] \ .$$

Against k-QR, k-SJS, k-$\mathsf{QR}^\star$, and k-$\mathsf{SJS}^\star$, its advantage is denoted

$$\alpha_1 := \left|\epsilon_1 - \tfrac{1}{4}(\epsilon_2' + \epsilon_2'') - \tfrac{1}{2}\epsilon_3\right|\,,\ \ \alpha_2 := \left|\tfrac{1}{2}(\epsilon_2' + \epsilon_2'') - \epsilon_3\right|\,,\ \ \alpha_3 := \left|\epsilon_1 - \tfrac{1}{2}(\epsilon_2' + \epsilon_2'')\right|\,,\ \text{ and }\ \alpha_4 := \left|\epsilon_2' - \epsilon_3\right|\,,$$

respectively.

We have to show that if the k-QR and k-SJS assumptions hold then so do the k-$\mathsf{QR}^\star$ and k-$\mathsf{SJS}^\star$ 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 $|\epsilon_2' - \epsilon_2''| \leq \alpha_1$.

From the definitions of α_3 and α_4, we can write

$$\begin{aligned}\alpha_3 &= \left|\epsilon_1 - \tfrac{1}{2}(\epsilon_2' + \epsilon_2'')\right| = \left|\epsilon_1 - \tfrac{1}{4}(\epsilon_2' + \epsilon_2'') - \tfrac{1}{2}\epsilon_3 + \tfrac{1}{2}\epsilon_3 - \tfrac{1}{4}(\epsilon_2' + \epsilon_2'')\right| \\ &\leq \left|\epsilon_1 - \tfrac{1}{4}(\epsilon_2' + \epsilon_2'') - \tfrac{1}{2}\epsilon_3\right| + \left|\tfrac{1}{2}\epsilon_3 - \tfrac{1}{4}(\epsilon_2' + \epsilon_2'')\right| \\ &= \alpha_1 + \tfrac{1}{2}\alpha_2\end{aligned}$$

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.

The lemma follows by noting that $\alpha_3 + \alpha_4 \leq \frac{3}{2}\alpha_1 + \frac{3}{2}\alpha_2 \leq 2(\alpha_1 + \alpha_2)$. $\square$

Theorem 1 (k-QR + k-SJS $\implies$ Gap-2^k-Res). *For RSA moduli* $N = pq$ *such that* $p, q \equiv 1 \pmod{2^k}$, *the Gap* 2^k*-Residuosity assumption holds if the* k-QR *assumption and the* k-SJS *assumption hold. More precisely, for any PPT distinguisher* $\mathcal{B}_0$ *against the latter, there exist* k-QR *distinguishers* $\mathcal{B}_1$ *and* $\mathcal{B}_3$ *and a* k-SJS *distinguisher* $\mathcal{B}_2$ *with comparable running times and for which*

$$\mathbf{Adv}_{\mathcal{B}_0}^{\mathsf{Gap}\text{-}2^k\text{-}\mathsf{Res}}(1^\kappa) \leq 2k \cdot \left(\mathbf{Adv}_{\mathcal{B}_1}^{k\text{-}\mathsf{QR}}(1^\kappa) + \mathbf{Adv}_{\mathcal{B}_2}^{k\text{-}\mathsf{SJS}}(1^\kappa)\right) + \mathbf{Adv}_{\mathcal{B}_3}^{k\text{-}\mathsf{QR}}(1^\kappa) \ .$$

Proof. To prove the result, we consider a sequence of distributions which will help us bridge the gap between the assumptions. More precisely, for $0 \leq i < k$, 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 another distribution 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 distributions of the corresponding subsets. Namely, unless either the k-$\mathsf{QR}^\star$ assumption or the k-$\mathsf{SJS}^\star$ assumption is false, we will prove

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

and, finally, that $D_{k-1} \overset{c}{\approx} R_k$ unless the k-QR assumption is false.

Note that since $D_{i-1} = \{y^{2^{i-1}} \mid y \in \mathbb{J}_N \setminus \mathbb{QR}_N\}$ and $D'_i = \{y^{2^i} \mid y \in \overline{\mathbb{J}}_N\} = \{x^{2^{i-1}} \mid x = y^2 \in \mathbb{QR}_N \text{ and } y \in \overline{\mathbb{J}}_N\}$, it turns out that D_{i-1} and D'_i are disjoint, for $1 \le i \le k-1$. The same is true for D'_i and D_i since $D'_i = \{y^{2^i} \mid y \in \overline{\mathbb{J}}_N\}$ and $D_i = \{y^{2^i} \mid y \in \mathbb{J}_N \setminus \mathbb{QR}_N\}$. Finally note that $D_{k-1} = \{y^{2^{k-1}} \mid y \in \mathbb{J}_N \setminus \mathbb{QR}_N\}$ and $R_k = \{y^{2^k} \mid y \in \mathbb{Z}_N^*\}$ are also disjoint.

Claim 1. If k-$\mathsf{QR}^\star$ holds, for each $i \in \{1, \dots, k-1\}$, no PPT adversary can distinguish the distributions of D_{i-1} and D'_i.

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-$\mathsf{QR}^\star$ distinguisher $\mathcal{B}_1$ with advantage ε for moduli $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$.

Our distinguisher $\mathcal{B}_1$ takes as input a composite integer $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$ and an element $w \in \mathbb{Z}_N^*$ which belongs to 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, $\mathcal{B}_1$ chooses a random element $z \xleftarrow{R} \overline{\mathbb{J}}_N$. 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, $\mathcal{B}_1$ outputs whatever $\mathcal{D}$ outputs.

- Let us first assume that $w \in \mathsf{dist}_0$. In this case, if w' denotes an arbitrary square root of w, we know that $w' \in \mathbb{J}_N$ and $x = (zw')^{2^i} \bmod N$. Further, since $z \xleftarrow{R} \overline{\mathbb{J}}_n$, we have $zw' \in \overline{\mathbb{J}}_N$ and thus $x \in_R D'_i$.
- Now let us 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-$\mathsf{SJS}^\star$ holds, for each $i \in \{1, \dots, k-1\}$, no PPT adversary can distinguish the distributions of D'_i and D_i.

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-$\mathsf{SJS}^\star$ distinguisher $\mathcal{B}_2$ with advantage ε for RSA moduli $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$. Given $w \in \mathbb{Z}_N^*$ which belongs to one of the two distributions

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

$\mathcal{B}_2$ constructs $x = w^{2^{i-1}} \bmod N$ which is used to feed the distinguisher $\mathcal{D}$. When the latter outputs a result, $\mathcal{B}_2$ produces the same output. It is clear that, if $w \in \mathsf{dist}_0$ (resp. $w \in \mathsf{dist}_1$), then x is in D_i (resp. D'_i). Hence, if $\mathcal{D}$ is a successful distinguisher, so is $\mathcal{B}_2$. ∎

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

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 $\mathcal{D}$.

Algorithm $\mathcal{B}_3$ takes as input $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$ 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 runs $\mathcal{D}$ on input of (x, k, N). When $\mathcal{D}$ halts and outputs $b \in \{0, 1\}$, $\mathcal{B}_1$ outputs the same bit.

It is easy to see that, if $w \in_R \mathbb{QR}_N$, then $x \in_R R_k$. If $w \in_R \mathbb{J}_N \setminus \mathbb{QR}_N$, we immediately have $x \in_R D_{k-1}$. ∎

To conclude the proof, we remark that, if a PPT distinguisher $\mathcal{B}_0$ exists for the Gap-2^k-Res assumption (i.e., if $D_0 \stackrel{c}{\not\approx} R_k$), then

- either $D_{k-1} \stackrel{c}{\not\approx} R_k$, contradicting k-QR (Claim 3); or
- there exists $1 \le i < k$ such that $D_i' \stackrel{c}{\not\approx} D_{i-1}$ or $D_i' \stackrel{c}{\not\approx} D_i$. 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. ☐

It is not hard to see that the semantic security of the scheme is equivalent to the Gap-2^k-Res assumption. We thus obtain the following theorem as a corollary.

Theorem 2. *The scheme is semantically secure under the k-QR and k-SJS assumptions. More precisely, for any IND-CPA adversary $\mathcal{A}$, we have either k-QR distinguishers $\mathcal{B}_1$ and $\mathcal{B}_3$ or an k-SJS distinguisher $\mathcal{B}_2$ such that*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A}}(1^\kappa) \le 2k \cdot \left(\mathbf{Adv}^{k\text{-}\mathsf{QR}}_{\mathcal{B}_1}(1^\kappa) + \mathbf{Adv}^{\mathsf{SJS}}_{\mathcal{B}_2}(1^\kappa)\right) + \mathbf{Adv}^{k\text{-}\mathsf{QR}}_{\mathcal{B}_3}(1^\kappa) \ .$$

Proof. 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 proof of Theorem 2 implicitly shows that, like the original Goldwasser-Micali system, our scheme is a *lossy* encryption scheme [4] (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 [16]. Moreover, for a lossy key (y, N), there exists an efficient algorithm that opens a given ciphertext c to any arbitrary plaintext m (by finding random coins that explain c as an encryption of m). It implies that our scheme satisfies the simulation-based definition [4] of selective-opening security.

4 Implementation and Performance

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

4.1 Parameter selection

The key generation (cf. § 3.1) requires two primes p and q such that $p, q \equiv 1 \pmod{2^k}$ and an 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 $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. So, we need to generate an element $y \in \mathbb{Z}_N^*$ such that *(i)* $y \bmod p$ is primitive in $\mathbb{Z}_p^*$, and *(ii)* $y \bmod q$ is primitive in $\mathbb{Z}_q^*$. Finding a primitive element modulo a prime number p is not difficult when the factorization of $p - 1$ is known. Therefore, we suggest to select prime p as a *k-quasi-safe prime*, that is, $p = 2^k p' + 1$ for some prime p' (likewise for prime q, we take $q = 2^k q' + 1$ for some prime q'). An efficient algorithm for generating k-quasi-safe primes is discussed in [29, Section 4.2].

Consider now the primitive 2^k-th root of unity $\zeta_{2^k} = e^{2i\pi/2^k}$ with $i = \sqrt{-1}$. It generates a cyclic group of order 2^k under multiplication. In our case, the key observation is that, when p is 2^k-quasi-safe prime, if y is a square modulo p then $\zeta_{2^k}\, y$ is not. Indeed, we have

$$\left(\frac{\zeta_{2^k}\, y}{p}\right) = \left(\frac{\zeta_{2^k}}{p}\right)\left(\frac{y}{p}\right) \equiv \zeta_{2^k}{}^{\frac{p-1}{2}}\left(\frac{y}{p}\right) \equiv (e^{i\pi})^{p'}\left(\frac{y}{p}\right) = -\left(\frac{y}{p}\right) \pmod{p}$$

since p' is odd. This leads to the following algorithm.

Algorithm 2 Generation of y

Input: Modulus $N = pq$ (with $p = 2^k p' + 1$ and $q = 2^k q' + 1$), primes p, q, p', q', and integer $k \geq 1$
Output: $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$

1: Pick at random $y_p \in \mathbb{Z}_p^*$ and $y_q \in \mathbb{Z}_q^*$
2: **if** $\left(\frac{y_p}{p}\right) = 1$ **then** $y_p \leftarrow \zeta_{2^k}\, y_p \bmod p$
3: **if** $\left(\frac{y_q}{q}\right) = 1$ **then** $y_q \leftarrow \zeta_{2^k}\, y_q \bmod q$
4: Set $y \leftarrow y_p + p\big(p^{-1}(y_q - y_p) \bmod q\big)$
5: **return** y

The primes p and q are chosen so that $p, q \equiv 1 \pmod{2^k}$. Sharing common factors for $(p-1)$ and $(q-1)$ was used already in several other systems; see e.g. [19,36]. Letting r denote a common factor of $(p-1)$ and $(q-1)$, a baby-step giant-step approach developed by McKee and Pinch [37] can factor RSA modulus $N = pq$ in essentially $O(N^{1/4}/r)$ operations. In our case, we have $r = 2^k$. For security it is therefore necessary that $\frac{1}{4}\log_2 N - k > \kappa$, or equivalently,

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

where κ is the security parameter.

A powerful LLL-based technique due to Coppersmith [13,14] also bounds the size of k to at most $\frac{1}{2}\min(\log_2 p, \log_2 q)$ bits as, otherwise, the factors of N would be revealed. Going beyond polynomial-time attacks, one should add an extra security margin to take into account exhaustive searches [42]. RSA moduli being balanced (i.e., $\frac{1}{2}\min(\log_2 p, \log_2 q) = \frac{1}{4}\log_2 N$), we so end up with the same upper bound as for the McKee-Pinch's approach: $k < \frac{1}{4}\log_2 N - \kappa$.

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, a specific parameter choice is $k = 128$ and $\log_2 N = 2048$.

4.2 Ciphertext expansion

Hybrid encryption allows designing efficient asymmetric schemes, as suggested by Shoup in the ISO 18033-2 standard for public-key encryption [27]. 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.

Table 1. Ciphertext expansion in a typical encryption

Encryption scheme	Assumption	Ciphertext size
Goldwasser-Micali [21]	Quadratic Residuosity (QR)	$k \cdot \log_2 N$
Benaloh-Fisher [12]	Prime residuosity (PR)	$\left\lceil \frac{k}{\log_2 r} \right\rceil \cdot \log_2 N$
Naccache-Stern [41]	Prime residuosity (PR)	$\log_2 N$
Okamoto-Uchiyama [44]	p-subgroup	$\log_2 N$
Paillier [45]	N-th residuosity	$2\log_2 N$
This paper	Quadratic residuosity (k-QR) Squared Jacobi Symbol (k-SJS)	$\log_2 N$

It appears that the Goldwasser-Micali cryptosystem has the higher ciphertext expansion but its semantic security relies on the standard quadratic residuosity assumption. The ciphertext expansion of Benaloh-Fischer cryptosystem is similar to that of 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. The Paillier cryptosystem produces twice larger ciphertexts.

The encryption scheme proposed in this paper has the same ciphertext expansion as in the Naccache-Stern cryptosystem. Moreover, its decryption algorithm is fast (it is even faster than in the Naccache-Stern cryptosystem), requires less memory, and the security relies on a quadratic residuosity assumption.

5 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 with much shorter outputs and keys than in previous QR-based or DDH-based examples.

In comparison with the function of Hemenway and Ostrovsky [23], for example, it compresses function values by a factor of k when we work with a modulus $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$. Moreover, the size of the evaluation key is decreased by a factor of $O(k^2)$ while increasing the lossiness by $2k$ more bits. Finally, our inversion trapdoor has constant size, whereas [23] 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.* [17,18], 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. Among known lossy trapdoor functions based on traditional number-theoretic assumptions [47,9,17,18,32,23,38], this appears as a rare efficiency tradeoff. To the best of our knowledge, it has only been achieved under the Composite Residuosity assumption [9,17,18] so far.

Interestingly, our LTDF provides similar efficiency improvements to the QR-based deterministic encryption scheme of Brakerski and Segev [11], which also builds on the Hemenway-Ostrovsky LTDF. Note that the scheme of [11] 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.

5.1 Description and security analysis

We start by recalling the following definition.

Definition 7 ([47]). *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* (InjGen, LossyGen, Eval, 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 $\ell_N(\kappa)$ and $k(\kappa)$ be security parameters determined by κ. Let also $n(\kappa)$ 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 a modulus $N = pq > 2^{\ell_N}$ such that $p = 2^k p' + 1$ and $q = 2^k q' + 1$ for primes p, q and odd prime integers p, q, p', q'. Choose $y \stackrel{R}{\leftarrow} \mathbb{J}_N \setminus \mathbb{QR}_N$.
2. For each $i \in \{1, \ldots, m\}$, pick h_i in the subgroup of order $p'q'$, by setting $h_i = g_i^{2^k} \bmod N$ for a randomly chosen $g_i \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$.
3. Choose $r_1, \ldots, r_m \stackrel{R}{\leftarrow} \mathbb{Z}_{p'q'}$ and compute a matrix $Z = \left(Z_{i,j}\right)_{i,j\in\{1,\ldots,m\}}$ 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 $ek := \left(N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}\right)$ and the trapdoor is $t := \{p, y\}$.

Sampling a lossy function. The process followed by $\mathsf{LossyGen}$ 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 $ek = \left(N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}\right)$, 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 $t = p$ and $\tilde{y} = (y_1, \ldots, y_m) \in \mathbb{Z}_N^m$, Invert applies the decryption algorithm of § 3.2 to each y_j, for $j = 1$ to m. Observe that when $(z_{ij})_{i,j\in\{1,\ldots,m\}}$ is the identity matrix, $\left(\frac{y_j}{p}\right)_{2^k} \equiv \left[\left(\frac{y}{p}\right)_{2^k}\right]^{x_j} \pmod p$. 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 [23] is slightly different in that, as in the DDH-based construction of Peikert and Waters [47], 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.

Theorem 3. *The above construction is a* $(n(\kappa), n(\kappa) - \log_2(p'q'))$*-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.*

Proof. 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 $y \stackrel{R}{\leftarrow} \mathbb{J}_N \setminus \mathbb{QR}_N$. The result of Theorem 1 shows that, under the $\mathsf{Gap}\text{-}2^k\text{-}\mathsf{Res}$ assumption, $\mathbf{Exp}_0$ is computationally indistinguishable from an experiment where the adversary is given the description of an injective function. Next, for each $i \in \{1, \ldots, m\}$ we define experiment $\mathbf{Exp}_i$ 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 columns all contain zeroes whereas the last $m-i$ columns are those of the $m \times m$ identity matrix.

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

The claim is proved in the same way as a similar claim about the DDH-based lossy TDF of Peikert and Waters [47]. Since y lives in the cyclic subgroup of 2^k-th residues, we are free to invoke the semantic security of the ElGamal encryption scheme (and thus the DDH assumption in this group) to justify that an ElGamal encryption of y can be replaced by an ElGamal encryption of 1 without any PPT distinguisher noticing. Concretely, the ElGamal challenger generates an ElGamal public key (g, h) in the subgroup of order $p'q'$. The public key is generated by setting $h_i = h$ and $h_j = g^{\alpha_j}$, with $\alpha_j \stackrel{R}{\leftarrow} \mathbb{Z}_{\lfloor N/4 \rfloor}$ for each $j \neq i$. We can define two messages $M_0 = y$ and $M_1 = 1$ and send them to the ElGamal challenger. The latter replies with a ciphertext $(C_0, C_1) = (g^r, M \cdot y^r)$ where either $M = y$ or $M = 1$. The evaluation key is generated by setting the entry (i, i) of the matrix as $Z_{i,i} = C_1$ while the i-th row is obtained by setting $Z_{i,j} = C_1^{\alpha_j}$. Other columns are generated by choosing the encryption exponents faithfully, as in Experiment $\mathbf{Exp}_{i-1}$. It should be clear that, if the ElGamal challenger chooses to encrypt y (resp. 1), the evaluation key is distributed as in Experiment $\mathbf{Exp}_{i-1}$ (resp. Experiment $\mathbf{Exp}_i$). □

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. □

It is worth noting that, with $N = pq$ such that $p, q \equiv 1 \pmod{2^k}$, a side effect of working in the subgroup of odd order is an improved lossiness. Indeed, we lose $n - \log_2(p'q')$ bits in comparison with $n - \log_2 \phi(N)$ in [23].

Using the techniques of Peikert and Waters [47], it is easy to construct an equally efficient all-but-one trapdoor function providing the same amount of lossiness under the same assumptions. A difference is that, in order to enable inversion, the resulting all-but-one function handles $k/2$ bits (instead of k) in each chunk. The details are given in Appendix A for completeness.

More importantly, the dimension m of the matrix and the output vector can be reduced to a fairly small constant, as illustrated below.

5.2 Efficiency

Here, we consider chosen-ciphertext security as the targeted application.

By combining the lossy and all-but-one trapdoor function, a CCA-secure encryption scheme can be obtained using the construction of [47]. We argue that $m = O(1)$ suffices for this purpose. Recall that the scheme of [47] 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 [15]. 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 $n - 2\log_2(p'q') \geq \nu$. If we set $k = \frac{1}{4}\log_2 N - \kappa$, we have $\log_2(p'q') = \log_2 \phi(N) - 2k \approx 4(k + \kappa) - 2k = 2k + 4\kappa$, which yields $n \geq 3k + 6\kappa$. If $k > \kappa$, it is sufficient to set

$n \geq 9k$. If we take into account the fact that our all-but-one function processes blocks of $k/2$ bits, we find that $m = 2n/k = 18$ suffices here.

As it turns out, when the Peikert-Waters construction [47, § 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 [25] (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.* [33], 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 [33] and additionally appeals to the k-SJS and DDH assumptions.

6 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

We are grateful to the anonymous reviewers of EUROCRYPT 2013 for their useful comments. We are also grateful to Fabrice Ben Hamouda for insightful discussions and to Javier Herranz for bringing [24] to our attention.

References

1. Michel Abdalla, Fabrice Ben Hamouda, and David Pointcheval. Tighter reductions for forward-secure signature schemes. In K. Kurosawa and G. Hanaoka, editors, *Public Key Cryptography – PKC 2013*, volume 7778 of *LNCS*, pages 292–311. Springer-Verlag, 2013.
2. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer-Verlag, 2007.
3. Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer-Verlag, 2009.
4. Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In A. Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer-Verlag, 2009.
5. Josh Daniel Cohen Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, New Haven, CT, USA, 1987.
6. Lenore Blum, Manuel Blum, and Mike Shub. Comparison of two pseudo-random number generators. In D. Chaum, R. L. Rivest, and A. T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 61–78. Plenum Press, 1983.
7. Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator. *SIAM J. Comput.*, 15(2):363–383, 1986.

8. Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology − CRYPTO '84*, volume 196 of *LNCS*, pages 289–302. Springer-Verlag, 1985.
9. Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *Advances in Cryptology − CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer-Verlag, 2008.
10. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *Theory of Cryptography Conference (TCC 2005)*, volume 3378 of *LNCS*, pages 325–341. Springer-Verlag, 2005.
11. Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In P. Rogaway, editor, *Advances in Cryptology − CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer-Verlag, 2001.
12. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme. In *26th Annual Symposium on Foundations of Computer Science (FOCS '85)*, pages 372–382. IEEE Computer Society, 1985.
13. Don Coppersmith. Finding a small root of a bivariate integer equation: Factoring with high bits known. In U. Maurer, editor, *Advances in Cryptology − EUROCRYPT '96*, volume 1070 of *LNCS*, pages 179–189. Springer-Verlag, 1996.
14. Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.
15. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology − EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.
16. Cynthia Dwork, Moni Naor, Omer Reingold, and Larry Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003.
17. David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography − PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer-Verlag, 2010.
18. David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 2012.
19. Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In I. B. Damgård, editor, *Advances in Cryptology − EUROCRYPT '90*, volume 473 of *LNCS*, pages 481–486. Springer-Verlag, 1991.
20. Oded Goldreich. *Foundations of Cryptography*, volume II. Cambridge University Press, 2004.
21. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
22. Jens Groth. Cryptography in subgroups of $\mathbb{Z}_n$. In J. Kilian, editor, *Theory of Cryptography Conference (TCC 2005)*, volume 3378 of *LNCS*, pages 50–65. Springer-Verlag, 2005.
23. Brett Hemenway and Rafail Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. Electronic Colloquium on Computational Complexity (ECCC), 2009.
24. Javier Herranz and Juan Ramón Sisternes. Cifrado homomórfico de clave pública basado en residuosidad cuadrática. RECSI 2012 − XII Reunión Española sobre Criptologìa y Seguridad de la Información, San Sebastián, 2012.
25. Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In A. Joux, editor, *Advances in Cryptology − EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer-Verlag, 2009.
26. Kenneth Ireland and Michael Rosen. *A Classical Introduction to Modern Number Theory*, volume 84 of *Graduate Texts in Mathematics*. Springer-Verlag, 2nd edition, 1990.
27. ISO/IEC 18033-2. Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers. International Organization for Standardization, May 2006.
28. Marc Joye and Benoît Libert. Efficient cryptosystems from 2^k-th power residue symbols. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology − EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 76–92. Springer-Verlag, 2013.
29. Marc Joye and Pascal Paillier. Fast generation of prime numbers on portable devices: An update. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems − CHES 2006*, volume 4249 of *LNCS*, pages 160–173. Springer-Verlag, 2006.
30. Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. CRC Press, 2007.
31. Jonathan Katz and Moti Yung. Threshold cryptosystems based on factoring. In Y. Zheng, editor, *Advances in Cryptology − ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 192–205. Springer-Verlag, 2002.
32. Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In T. Rabin, editor, *Advances in Cryptology − CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313. Springer-Verlag, 2010.

33. Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In A. Joux, editor, *Advances in Cryptology − EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 590–609. Springer-Verlag, 2009.
34. Kaoru Kurosawa, Yutaka Katayama, Wakaha Ogata, and Shigeo Tsujii. General public key residue cryptosytems and mental poker protocols. In I. B. Damgård, editor, *Advances in Cryptology − EUROCRYPT '90*, volume 473 of *LNCS*, pages 374–388. Springer-Verlag, 1991.
35. Franz Lemmermeyer. *Reciprocity Laws*. Springer Monographs in Mathematics. Springer-Verlag, 2000.
36. Chae Hoon Lim and Pil Joong Lee. Security and performance of served-aided RSA computation protocols. In D. Coppersmith, editor, *Advances in Cryptology − CRYPTO '95*, volume 963 of *LNCS*, pages 70–83. Springer-Verlag, 1995.
37. James McKee and Richard Pinch. Further attacks on server-aided RSA cryptosystems. Unpublished manuscript, 1998.
38. Petros Mol and Scott Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography − PKC 2010*, volume 6056 of *LNCS*, pages 296–311. Springer-Verlag, 2010.
39. Jean Monnerat and Serge Vaudenay. Generic homomorphic undeniable signatures. In P. J. Lee, editor, *Advances in Cryptology − ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 354–371. Springer-Verlag, 2004.
40. Jean Monnerat and Serge Vaudenay. Undeniable signatures based on characters: How to sign with one bit. In F. Bao et al., editors, *Public Key Cryptography − PKC 2004*, volume 2947 of *LNCS*, pages 69–75. Springer-Verlag, 2004.
41. David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *ACM Conference on Computer and Communications Security 1998 (CCS '98)*, pages 59–66. ACM Press, 1998.
42. Phong Q. Nguyen. Public-key cryptanalysis. In I. Luengo, editor, *Recent Trends in Cryptography*, Contemporary Mathematics. AMS–RSME, 2009.
43. Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In K. Kim, editor, *Public Key Cryptography (PKC 2001)*, volume 1992 of *LNCS*, pages 308–318. Springer-Verlag, 2001.
44. Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In K. Nyberg, editor, *Advances in Cryptology − EUROCRYPT '98*, volume 1403 of *LNCS*, pages 308–318. Springer-Verlag, 1998.
45. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology − EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer-Verlag, 1999.
46. Sung Jun Park, Bo Young Lee, and Dong Ho Won. A probabilistic encryption using very high residuosity and its applications. In *Global Telecommunications Conference (GLOBECOM '95)*, pages 1179–1182. IEEE Press, 1995.
47. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In C. Dwork, editor, *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, pages 187–196. ACM Press, 2008.
48. Stephen H. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Tran. Inf. Theory*, 24(1):106–110, 1978.
49. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. Earlier version in STOC 2005.
50. Renate Scheidler. A public-key cryptosystem using purely cubic fields. *J. Cryptology*, 11(2):109–124, 1998.
51. Renate Scheidler and Hugh C. Williams. A public-key cryptosystem utilizing cyclotomic fields. *Des. Codes Cryptography*, 6(2):117–131, 1995.
52. Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2nd edition, 2010.
53. Hoeteck Wee. Dual projective hashing and its applications - Lossy trapdoor functions and more. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology − EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer-Verlag, 2012.
54. Song Y. Yan. *Number Theory for Computing*. Springer-Verlag, 2nd edition, 2002.
55. Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. Residuosity problem and its applications to cryptography. *Trans. IEICE*, E-71(8):759–767, 1988.

A 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 ($\mathsf{BranchGen}, \mathsf{ABOGen}, \mathsf{Eval}, \mathsf{Invert}$) with the following specifications.

– *Sampling a branch:* Given a security parameter κ, $\mathsf{BranchGen}$ 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 $\mathsf{BranchGen}$. It outputs the description ek of a function and a trapdoor t.
– *Evaluation:* For any branch $b^\star$ produced by $\mathsf{BranchGen}$, 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 $\mathsf{BranchGen}$ 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 $\mathsf{BranchGen}$, 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 [17] 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 another security parameter $\lambda(\kappa)$ 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, other security parameters $\ell_N(\kappa)$ and $\lambda(\kappa)$ that are determined by κ, the desired input length $n(\kappa)$ 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 RSA modulus $N = pq > 2^{\ell_N}$ such that $p = 2^k p' + 1$ and $q = 2^k q' + 1$ for large primes p, q and odd prime integers p', q'. Choose $y \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$.
2. For each $i \in \{1, \ldots, m\}$, pick h_i in the subgroup 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

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

where $\left(z_{i,j}\right)_{i,j\in\{1,\ldots,m\}}$ 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 $ek := \left(N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}\right)$ 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 $ek = \left(N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}\right)$, 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 $ek = \Big(N, (Z_{i,j})_{i,j\in\{1,\ldots,m\}}\Big)$ 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 § 3.2 to w_i.
3. From the vector of plaintexts $\tilde{x} = (x_1, \ldots, x_m) \in \mathbb{Z}^m_{2^\lambda}$ obtained at Step 2, define $\tilde{x}' = (x'_1, \ldots, x'_m) \in \mathbb{Z}^m_{2^\lambda}$ such that $x'_i = x_i/\mathrm{abs}(b - b^\star)$ (the division being performed over $\mathbb{Z}$), where $\mathrm{abs}(b - b^\star) = b - b^\star$ 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 $\mathrm{abs}(b - b^\star) \cdot x_i < 2^{2\lambda} = 2^k$ 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 and if the DDH assumption holds in the subgroup of odd order. The proof is essentially identical to the proof of Theorem 3 and omitted.

B Optimized Decryption Algorithms

In its basic version, the decryption requires $O(k)$ modular exponentiations in $\mathbb{Z}_p^*$ in order to compute higher power residue symbols. As a result, the number of bit operations is quartic in the security parameter. This section shows that a suitable pre-processing phase allows significantly increasing the decryption speed. At the price of extra storage requirements, the receiver's workload even drops to a cubic number of bit operations, which is asymptotically on par with, e.g., Paillier and Okamoto-Uchiyama cryptosystems.

B.1 First modified decryption algorithm

In the following, we define the private decryption key as (p', D) for some precomputed value D.

Let $N = pq$ where p and q are prime, and $p, q \equiv 1 \pmod{2^k}$ but $p, q \not\equiv 1 \pmod{2^{k+1}}$. In this case, we can write $p = 2^k p' + 1$ and $q = 2^k q' + 1$ for some odd integers p' and q'.

Therefore as $p = 2^k p' + 1$ with p' odd and since $y \in \mathbb{J}_N \setminus \mathbb{QR}_N$, it is easily seen that

$$y^{2^{k-1}p'} \equiv -1 \pmod{p}\ .$$

The proof of this statement is straightforward since we have $y^{2^{k-1}p'} \equiv y^{(p-1)/2} \equiv \left(\frac{y}{p}\right) \equiv -1 \pmod{p}$.

Now, 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 $\lambda_j = 2^{k-j}p'$ and

$$C_j = c^{\lambda_j} \bmod p\ .$$

Then, we have

$$\begin{aligned}C_j &\equiv \Big(y^m x^{2^k}\Big)^{\lambda_j} \equiv y^{m2^{k-j}p'} \equiv y^{(m \bmod 2^j)2^{k-j}p'} \equiv y^{(m_{j-1}2^{j-1}+\sum_{i=0}^{j-2} m_i 2^i)2^{k-j}p'} \equiv y^{m_{j-1}2^{k-1}p'}\, y^{(m \bmod 2^{j-1})2^{k-j}p'}\\ &\equiv (-1)^{m_{j-1}}\, y^{(m \bmod 2^{j-1})\lambda_j} \pmod{p}\ .\end{aligned}$$

Hence, it follows that

$$\left(\frac{c}{y^{m \bmod 2^{j-1}}}\right)^{\lambda_j} \equiv (-1)^{m_{j-1}} \pmod{p} \ .$$

This yields the following decryption algorithm. The private key now consists of the pair (p', D) where $D = y^{-p'} \bmod p$. The input is the ciphertext c and the output is plaintext m.

$\mathsf{m} \leftarrow 0$; $\mathsf{B} \leftarrow 1$; $\mathsf{D} \leftarrow D$
$\mathsf{C} \leftarrow c^{p'} \bmod p$
for $j = 1$ to k **do**
  $\mathsf{z} \leftarrow (\mathsf{C} \cdot \mathsf{D}^{\mathsf{m}})^{2^{k-j}} \bmod p$
  if $(\mathsf{z} \neq 1)$ **then** $\mathsf{m} \leftarrow \mathsf{m} + \mathsf{B}$
  $\mathsf{B} \leftarrow 2\mathsf{B}$
end for
return m

Variable m in the for-loop contains the lowest part of the plaintext m. As one bit of plaintext m is correctly obtained per iteration, there is no need to recompute $D^{\mathsf{m}} \bmod p$. Instead, it suffices to update it using C as an accumulator. Further, we only perform the for-loop until iteration $k-1$ to save a couple of operations. We thus obtain:

Algorithm 3 New decryption algorithm

Input: Ciphertext c, private key (p', D) (and public-key elements y and k)
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

$\mathsf{m} \leftarrow 0$; $\mathsf{B} \leftarrow 1$; $\mathsf{D} \leftarrow D$
$\mathsf{C} \leftarrow c^{p'} \bmod p$
for $j = 1$ to $k-1$ **do**
  $\mathsf{z} \leftarrow \mathsf{C}^{2^{k-j}} \bmod p$
  if $(\mathsf{z} \neq 1)$ **then** $\mathsf{m} \leftarrow \mathsf{m} + \mathsf{B}$; $\mathsf{C} \leftarrow \mathsf{C} \cdot \mathsf{D} \bmod p$
  $\mathsf{B} \leftarrow 2\mathsf{B}$; $\mathsf{D} \leftarrow \mathsf{D}^2 \bmod p$
end for
if $(\mathsf{C} \neq 1)$ **then** $\mathsf{m} \leftarrow \mathsf{m} + B$
return m

As a variant, we remark that D can be precomputed instead as being explicitly included in the private key.

B.2 Second modified decryption algorithm

Compared to the basic decryption algorithm (Alg. 1), the new decryption algorithm (Alg. 3) does not require more memory for its implementation. We present below a even faster variant by augmenting the private key.

We now define the decryption key to be the tuple $(p', D[1], \ldots, D[k-1])$ where $D[j] = D^{2^{j-1}} \bmod p$ for $1 \leq j \leq k-1$.

Algorithm 4 New decryption algorithm (2)

Input: Ciphertext c, private key $(p', D[1], \ldots, D[k-1])$ (and public-key elements y and k)
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

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

B.3 Third modified decryption algorithm

Yet a further optimization consists in computing the different values of $\mathsf{C}^{2^{k-j}}$ before the for-loop. We start with our first algorithm. The decryption key is now the pair $(p', \tilde{D})$ where $\tilde{D} = D^{-1} \bmod p$; i.e., $\tilde{D} = y^{p'} \bmod p$.

Algorithm 5 New decryption algorithm (3)

Input: Ciphertext c, private key $(p', \tilde{D})$ (and public-key elements y and k)
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

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

We observe that, at the cost of storing $\{\mathsf{C}[j]\}_{j=0}^{k-1}$ from step 3 to step 7, the decryption complexity is now dominated by that of one exponentiation in $\mathbb{Z}_p^*$ as well as $2k$ squarings and k multiplications, which incurs $O(k^3)$ bit operations.

B.4 Fourth modified decryption algorithm

We apply the modification as in our second decryption algorithm to the previous algorithm. The decryption key is defined as the tuple $(p', \tilde{D}[1], \ldots, \tilde{D}[k-1])$ where $\tilde{D}[j] = \tilde{D}^{2^{j-1}} \bmod p$ for $1 \leq j \leq k-1$.

Algorithm 6 New decryption algorithm (4)

Input: Ciphertext c, private key $(p', \tilde{D}[1], \ldots, \tilde{D}[k-1])$ (and public-key elements y and k)
Output: Plaintext $m = (m_{k-1}, \ldots, m_0)_2$

$\mathsf{m} \leftarrow 0$; $\mathsf{A} \leftarrow 1$; $\mathsf{B} \leftarrow 1$
for $j = 1$ to $k-1$ **do** $\mathsf{D}[j] \leftarrow \tilde{D}[j]$
$\mathsf{C}[0] \leftarrow c^{p'} \bmod p$
for $j = 1$ to $k-1$ **do** $\mathsf{C}[j] \leftarrow \mathsf{C}[j-1]^2 \bmod p$
for $j = 1$ to $k-1$ **do**
    if $(\mathsf{A} \neq \mathsf{C}[k-j])$ **then** $\mathsf{m} \leftarrow \mathsf{m} + \mathsf{B}$; $\mathsf{A} \leftarrow \mathsf{A} \cdot \mathsf{D}[j] \bmod p$
    $\mathsf{B} \leftarrow 2\mathsf{B}$
end for
if $(\mathsf{A} \neq \mathsf{C}[0])$ **then** $\mathsf{m} \leftarrow \mathsf{m} + B$
return m