Threshold Cryptosystems Based on 2^k-th Power Residue Symbols

George Teşeleanu[1,2]

[1] Advanced Technologies Institute
10 Dinu Vintilă, Bucharest, Romania
[2] Simion Stoilow Institute of Mathematics of the Romanian Academy
21 Calea Grivitei, Bucharest, Romania
george.teseleanu@yahoo.com

Abstract. In this paper we introduce a novel version of the Joye-Libert cryptosystem that allows users to decrypt without knowing the factorisation of the composite modulus. Then we use our construction as a building block for a threshold decryption protocol of the homomorphic Joye-Libert encryption scheme. Finally, we present several extensions of the threshold cryptosystem.

1 Introduction

In classical public key encryption systems, only the owner of the secret key has the ability to decrypt ciphertexts. Unfortunately, if an adversary is able to break into a system administrator's computer, for example, and steal its secret key, the whole system is compromised. Since, this type of attack by hackers or Trojan horses or corrupted insiders becomes more frequent and more easily to perform, the need to develop a method of distributing trust arises. In order to address this issue, a possible solution is to distribute the secret key between several servers and then use threshold decryption algorithms.

Most previous research has mainly focused on developing threshold decryption algorithms for RSA-based schemes [5,11,16,17] and discrete logarithm-based schemes [6, 12, 16, 32, 38]. But according to [9, 17, 26], there is still a need to design threshold schemes for many specific cryptosystems. Furthermore, as many have pointed out previously [9,11,17,18,25], threshold homomorphic schemes are useful for achieving goals such as electronic voting and efficient multi-party computation. In line with this reasoning, Katz and Yung [26] developed a threshold cryptosystem based on the Goldwasser-Micali encryption scheme [21, 22]. Moreover, their conversion keeps the homomorphic properties of the original scheme. The Katz-Yung scheme is revisited in [13] with the goal of extending it to composite moduli for which the Katz-Yung scheme fails.

A rather natural extension of the Goldwasser-Micali cryptosystem was introduced by Joye and Libert in [23] and it was reconsidered in [4]. Despite the fact that it is simple and elegant, the Goldwasser-Micali scheme is quite uneconom-

ical in terms of bandwidth[3]. Various attempts of generalizing the Goldwasser-Micali scheme were proposed in the literature in order to address the previously mentioned issue. The Joye-Libert scheme can be considered a follow-up of the cryptosystems proposed in [7,29] and efficiently supports the encryption of larger messages. The authors of [23] leave as an open problem the extension of their scheme, starting from [26], to a threshold decryption scheme.

Having in mind the motivations stated in the previous paragraphs, in this paper we develop a threshold version of the Joye-Libert cryptosystem [4, 23] that generalizes both the Katz-Yung scheme [26] and the Desmedt-Kurasawa scheme [13]. Note that our generalization conserves the homomorphic property of the Joye-Libert cryptosystem.

Another important problem that we address is proving the security of our threshold decryption scheme against chosen ciphertexts attacks. This topic was tackled by Katz and Yung for their scheme [26]. More precisely, they applied the generic conversion method from [16] that uses two independent encryption runs and a non-interactive zero-knowledge proof that the resulting ciphertexts contain the same message. Although, Katz and Yung provide such a proof system, they do not formally prove it secure. On the other hand, Desmedt and Kurasawa [13] simply state that proving the chosen ciphertexts security for their scheme is quite complex, and thus they only focus on semantic security. Therefore, we wanted to fill these gaps. When we tried to directly generalize Katz and Yung's proof, we ended up with a cumbersome protocol. Hence, starting from the examples described in [16] and the signature protocol from [20], we constructed a novel non-interactive zero-knowledge proof that is suitable for our threshold scheme and then we prove it secure. Note that our proof is also suitable for the Katz-Yung and Desmedt-Kurasawa schemes.

Structure of the paper. In Section 2 we introduce notations, definitions, security assumptions and schemes used throughout the paper. Inspired by the Joye-Libert encryption scheme, in Section 3 we propose a new scheme based on 2^k residues, prove it secure in the standard model and analyze its performance compared to other related cryptosystems. A threshold version of our scheme is proposed in Section 4 and extensions are given in Section 5. We conclude in Section 6.

2 Preliminaries

Notations. Throughout the paper, λ denotes a security parameter. We use the notation $x \xleftarrow{\$} X$ when selecting a random element x from a sample space X. We denote by $x \leftarrow y$ the assignment of the value y to the variable x. The probability that event E happens is denoted by $Pr[E]$.

The Jacobi symbol of an integer a modulo an integer n is generally represented by $\left(\frac{a}{n}\right)$. J_n and $\bar{J}_n$ denote the sets of integers modulo n with Jacobi

[3] $k \cdot \log_2 n$ bits are needed to encrypt a k-bit message, where n is a composite modulus as described in [21, 22]

symbol 1, respectively -1. Throughout the paper, we let QR_n be the set of quadratic residues modulo n. We define the alternative representation of integers modulo an integer p as $\mathcal{Z}_p = \{-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2\}$. The set of integers $\{0, \ldots, a-1\}$ is further denoted by $[0, a)$. For shorthand, we denote the set $[0, a+1)$ by $[0, a]$. Multidimensional vectors $v = (v_0, \ldots, v_{s-1})$ are represented as $v = \{v_i\}_{i \in [0,s)}$.

2.1 Number Theoretic Prerequisites

The Legendre symbol can be generalized to higher powers in several ways. We further consider the 2^k-th power residue symbol as presented in [40]. The classical Legendre symbol is obtained when $k = 1$.

Definition 1. *Let p be an odd prime such that $2^k | p-1$. Then the symbol*

$$\left(\frac{a}{p}\right)_{2^k} = a^{\frac{p-1}{2^k}} \pmod{p}$$

is called the 2^k-th power residue symbol modulo p, where $a^{\frac{p-1}{2^k}} \in \mathcal{Z}_p$.

Properties. The 2^k-th power residue symbol satisfies the following properties

1. If $a \equiv b \pmod{p}$, then $\left(\frac{a}{p}\right)_{2^k} = \left(\frac{b}{p}\right)_{2^k}$,
2. $\left(\frac{a^{2^k}}{p}\right)_{2^k} = 1$,
3. $\left(\frac{ab}{p}\right)_{2^k} = \left(\frac{a}{p}\right)_{2^k}\left(\frac{b}{p}\right)_{2^k} \pmod{p}$,
4. $\left(\frac{1}{p}\right)_{2^k} = 1$ and $\left(\frac{-1}{p}\right)_{2^k} = (-1)^{(p-1)/2^k}$.

In our paper we will make use of a generalized version of the Chinese Remainder Theorem. More precisely, we are interested in the case of moduli that are not pairwise coprime. We further present the theorem as stated in [33].

Theorem 1 (Generalized Chinese Remainder Theorem). *Let $m_1, m_2, \ldots, m_t$ be positive integers. For a set of integers $a_1, a_2, \ldots, a_t$ the system of congruences*

$$x \equiv a_i \pmod{m_i}, \text{ for } i \in [1, t]$$

has solutions if and only if

$$a_i \equiv a_j \pmod{\gcd(m_i, m_j)}, \text{ for } i \neq j, \ i, j \in [1, t]. \tag{1}$$

If Equation (1) holds, then the solution will be unique modulo $\operatorname{lcm}(m_1, m_2, \ldots, m_t)$.

We additionally use a theorem proved by Dirichlet in 1837. This theorem establishes the constrains necessary for the existence of infinitely many primes in an arithmetic progression. The original proof can be found in [27].

Theorem 2 (Dirichlet's theorem). *Let r, q be two coprime positive integers and let $\{a_n\}_{n\in\mathbb{N}}$ be an arithmetic progression such that $a_n = qn + r$. Then there exists a subsequence $\{b_{n'}\}_{n'\in\mathbb{N}} \subseteq \{a_n\}_{n\in\mathbb{N}}$ such that $b_{n'}$ is prime for each n'.*

2.2 Computational Complexity

To analyze the performance of our scheme, we must consider the complexities of the mathematical operations listed in Table 1. These complexities are in line with those presented in [10]. Note that, instead of using the explicit formula for the complexity of multiplication, we simply denote it by $M(\cdot)$.

Table 1. Computational complexity for μ-bit numbers and k-bit exponents

Operation	Complexity
Multiplication	$M(\mu) = \mathcal{O}(\mu \log(\mu) \log(\log(\mu)))$
Exponentiation	$\mathcal{O}(kM(\mu))$
Jacobi symbol	$\mathcal{O}(\log(\mu)M(\mu))$

2.3 Security Assumptions

Definition 2 (Quadratic Residuosity - QR, Squared Jacobi Symbol - SJS and Gap 2^k-Residuosity - GR). *Choose two large prime numbers $p, q \geq 2^\lambda$ and compute $n = pq$. Let A be a probabilistic polynomial-time (PPT) algorithm that returns 1 on input (x, n) or (x^2, n) or (x, k, n) if $x \in QR_n$ or J_n or $J_n \backslash QR_n$. We define the advantages*

$$ADV_A^{\mathrm{QR}}(\lambda) = \left| Pr[A(x,n) = 1 | x \xleftarrow{\$} QR_n] - Pr[A(x,n) = 1 | x \xleftarrow{\$} J_n \setminus QR_n] \right|,$$
$$ADV_A^{\mathrm{SJS}}(\lambda) = \left| Pr[A(x^2,n) = 1 | x \xleftarrow{\$} J_n] - Pr[A(x^2,n) = 1 | x \xleftarrow{\$} \bar{J}_n] \right|,$$
$$ADV_{A,k}^{\mathrm{GR}}(\lambda) = \left| Pr[A(x,k,n) = 1 | x \xleftarrow{\$} J_n \setminus QR_n] - Pr[A(x^{2^k},k,n) = 1 | x \xleftarrow{\$} \mathbb{Z}_n^*] \right|.$$

The Quadratic Residuosity assumption states that for any PPT algorithm A the advantage $ADV_A^{\mathrm{QR}}(\lambda)$ is negligible.

If $p, q \equiv 1 \bmod 4$, then the Squared Jacobi Symbol assumption states that for any PPT algorithm A the advantage $ADV_A^{\mathrm{SJS}}(\lambda)$ is negligible.

Let $p, q \equiv 1 \bmod 2^k$. The Gap 2^k-Residuosity assumption states that for any PPT algorithm A the advantage $ADV_A^{\mathrm{GR}}(\lambda)$ is negligible.

Remark 1. In [4], the authors investigate the relation between the assumptions presented in Definition 2. They prove that for any PPT adversary A against the GR assumption, we have two efficient PPT algorithms B_1 and B_2 such that

$$ADV^{\text{GR}}_{A,k}(\lambda) \leq \frac{3}{2}\left((k-\frac{1}{3})\cdot ADV^{\text{QR}}_{B_1}(\lambda) + (k-1)\cdot ADV^{\text{SJS}}_{B_2}(\lambda)\right).$$

Note that the QR assumption is a well studied security assumption, while the GR assumption is a relatively new one. Therefore, the relationship between these assumptions provides us with an additional level of security assurance.

2.4 Public Key Encryption

A *public key encryption* (PKE) scheme usually consists of three PPT algorithms: *Setup*, *Encrypt* and *Decrypt*. The *Setup* algorithm takes as input a security parameter and outputs the public key as well as the matching secret key. *Encrypt* takes as input the public key and a message and outputs the corresponding ciphertext. The *Decrypt* algorithm takes as input the secret key and a ciphertext and outputs either a valid message or an invalidity symbol (if the decryption failed).

Definition 3 (Indistinguishability under Chosen Plaintext Attacks - IND-CPA). *The security model against chosen plaintext attacks for a PKE scheme is captured in the following game:*

- *Setup(λ): The challenger C generates the public key, sends it to adversary A and keeps the matching secret key to himself.*
- *Query: Adversary A sends to C two equal length messages m_0, m_1. The challenger flips a coin $b \in \{0,1\}$ and encrypts m_b. The resulting ciphertext c is sent to the adversary.*
- *Guess: In this phase, the adversary outputs a guess $b' \in \{0,1\}$. He wins the game, if $b' = b$.*

The advantage of an adversary A attacking a PKE scheme is defined as

$$ADV^{\text{IND-CPA}}_{A}(\lambda) = |Pr[b = b'] - 1/2|$$

where the probability is computed over the random bits used by C and A. A PKE scheme is IND-CPA *secure, if for any PPT adversary A the advantage $ADV^{\text{IND-CPA}}_{A}(\lambda)$ is negligible.*

Definition 4 (Indistinguishability under Chosen Ciphertext Attacks - IND-CCA). *In the context of Definition 3, if before and after the query phase the adversary has access to a decryption oracle, we say that scheme is* IND-CCA *secure. The only restriction imposed on the adversary is that after the query phase he cannot query the decryption oracle with input c.*

The Joye-Libert PKE scheme. The Joye-Libert scheme was introduced in [23] as a generalization of the Goldwasser-Micali cryptosystem [21] to multi-bit messages. The scheme is proven secure in the standard model under the GR assumption [4, 23]. We shortly describe the algorithms of the Joye-Libert cryptosystem.

Setup(λ): Set an integer $k \geq 1$. Randomly generate two distinct large prime numbers p, q such that $p, q \geq 2^{\lambda}$ and $p, q \equiv 1 \bmod 2^k$. Output the public key $pk = (n, y, k)$, where $n = pq$ and $y \in J_n \setminus QR_n$. The corresponding secret key is $sk = (p, q)$.

Encrypt(pk, m): To encrypt a message $m \in [0, 2^k)$, we choose $x \xleftarrow{\$} \mathbb{Z}_n^*$ and compute $c \equiv y^m x^{2^k} \bmod n$. Output the ciphertext c.

Decrypt(sk, c): Compute $z \equiv \left(\frac{c}{p}\right)_{2^k}$ and find m such that the relation $\left[\left(\frac{y}{p}\right)_{2^k}\right]^m \equiv z \bmod p$ holds. Efficient methods to recover m can be found in [24].

Threshold PKE schemes. Compared to PKE schemes, the *Setup* and *Decrypt* algorithms of threshold schemes use sub-algorithms to distribute/aggregate information to/from participants. More precisely, the *Setup* algorithm takes as input a security parameter, the number of total players ℓ and the decryption threshold h; it outputs the public key and distributes the shares of the secret key to the ℓ players. The *Decrypt* algorithm takes as input a ciphertext; it forwards it to player i's decryption algorithm[4]; aggregates the decryption shares from each player and after receiving at least h shares it outputs either a valid message or an invalidity symbol.

In our paper we will consider the definition of a simulatable threshold protocol introduced by Gennaro *et al.* in [19]. Informally, a protocol is simulatable if we can show how an adversary attacking the original scheme can simulate the view of $h - 1$ players. This implies that this adversary can use an efficient attacker against the threshold version to break the original protocol. Hence, we show that if the original PKE is IND-CPA secure and the threshold version is simulatable, then the threshold PKE is IND-CPA secure even when the adversary has corrupted $h - 1$ players.

2.5 Non-Interactive Zero-Knowledge Protocols

Let $Q : \{0,1\}^* \times \{0,1\}^* \to \{\texttt{true}, \texttt{false}\}$ be a predicate and let $\mathcal{L} \in NP$ be a language. Given a value $z \in \mathcal{L}$, Peggy will try to convince Victor that she knows a value ω such that $Q(z, \omega) = \texttt{true}$.

We further base our reasoning on the definitions from [15, 30, 37] which we recall next.

Definition 5 (Non-Interactive Proof of Knowledge Protocol). *A protocol (f, P, V) is a non-interactive proof of knowledge protocol for predicate Q*

[4] which has access to player i's secret key share

if f is a polynomial and P and V are PPT algorithms such that the following properties hold

- **Completeness***: For all $z \in \mathcal{L}$, all ω such that $Q(z, \omega) = \texttt{true}$ and all strings σ of length $f(|x|)$, on input $(x, P(x, \omega, \sigma), \sigma)$ V accepts P's proof;*
- **Soundness***: For any adversary $\bar{P}$, if $\sigma \xleftarrow{\$} \{0,1\}^{f(k)}$ is chosen randomly, then the probability of $\bar{P}$ outputting an (x, p) such that $x \notin \mathcal{L}$ and on input (x, p, σ) V accepts $\bar{P}$'s proof is negligible in k.*

Definition 6 (Non-Interactive Zero Knowledge Protocol). *A protocol (f, P, V) is zero-knowledge if for every efficient program $\bar{V}$ there exists an efficient program S, the simulator, such that the output of S is indistinguishable from a transcript of the protocol execution between P and $\bar{V}$. If the indistinguishability is perfect*[5]*, then the protocol is called perfect zero-knowledge.*

The soundness property of a non-interactive proof of knowledge protocol (NIZK) states that with overwhelming probability, the prover P should be incapable of convincing the verifier V of a false statement. In the following definition, we consider a stronger notion, namely that this remains the case even after a polynomially bounded party has seen a simulated proof of its choosing.

Definition 7 (Simulation-Soundness). *A protocol (f, P, V) is simulation-sound if for every efficient program $\bar{P}$ there exists an efficient program S, the simulator, such that for any bounded list (Σ, κ) of proven words produced by S, any word z computed by $\bar{P}(\Sigma)$ and any proof p calculated by $S(x, \Sigma, \kappa)$, the probability of $\bar{P}(x, p, \Sigma, \tau)$ outputting an (x', p') such that $p' \neq p$, $x' \notin \mathcal{L}$ and on input (x', p', Σ) V accepts $\bar{P}$'s proof is negligible in k.*

Twin-Encryption Construction Based on the work of Naor and Yung [30], Fouque and Pointcheval [16] describe a generic method for converting IND-CPA secure PKEs into IND-CCA secure ones. One of the building block of their construction is a NIZK that convinces everybody that two encryption contain the same message. In order to show that the proof used in their construction is simulation-sound, we first need to define our language

$$\mathcal{L}_{pk_0, pk_1} = \{(\mathcal{E}_{pk_0}(m), \mathcal{E}_{pk_1}(m)) \mid \text{for any message } m\},$$

where $\mathcal{E}_{pk_i}(m)$ is the encryption of m with the public key pk_i, for $i \in [0, 1]$. If a pair (a_0, a_1) is a member of the language $\mathcal{L}_{pk_0, pk_1}$ we say that the pair is valid. Otherwise, the pair is invalid.

Using this type of NIZK, Fouque and Pointcheval [16] provide two conversion examples. One for the ElGamal cryptosystem [14] and one for Paillier cryptosytem [31]. Similarly, Katz and Yung [26] provide a NIZK for the Goldwasser-Micalli cryptosystem [21] when both primes are congruent with 3 modulo 4. Note that the only NIZK proven to be simulation-sound is the one used for converting the ElGamal PKE. For the remaining ones the proofs are omitted.

[5] i.e. the probability distribution of the simulated and the actual transcript are identical

3 A Public Key Encryption Scheme

3.1 Prerequisites

Lemma 1. *Let $k, \alpha > 0$ be integers and let $s \in \mathbb{Z}_{2^\alpha}$ be odd. For a pair of distinct prime numbers p, q such that*

$$p \equiv q \equiv s \cdot (2^\alpha)^k + 1 \pmod{(2^\alpha)^{k+1}},$$

we have

$$\gcd(p-1, q-1) | (p-q)/2^\alpha.$$

Proof. We first remark that from the definition of p and q we obtain $2^\alpha | p - q$.

Lets consider an odd integer r such that $r | \gcd(p-1, q-1)$. In this case, we obtain that $r | p - q$ and taking into account the property $\gcd(2, r) = 1$ we derive the relation $r | (p-q)/2^\alpha$.

We further examine the power of 2 in the prime factorization of the integer $\gcd(p-1, q-1)$. According to the definition we have

$$\begin{aligned} p &= p' \cdot (2^\alpha)^{k+1} + s \cdot (2^\alpha)^k + 1, \\ q &= q' \cdot (2^\alpha)^{k+1} + s \cdot (2^\alpha)^k + 1, \end{aligned}$$

where p', q' are positive integers. Hence, we obtain that

$$p - q = (p' - q')(2^\alpha)^{k+1}. \tag{2}$$

Since s is odd, we have that

$$2^{\alpha k} | p-1, 2^{\alpha k+1} \not| p-1 \text{ and } 2^{\alpha k} | q-1, 2^{\alpha k+1} \not| q-1,$$

and thus

$$2^{\alpha k} | \gcd(p-1, q-1) \text{ and } 2^{\alpha k+1} \not| \gcd(p-1, q-1).$$

In consequence, we need to show that $2^{\alpha k} | (p-q)/2^\alpha$, or equivalently that $2^{\alpha(k+1)} | p - q$. But this is true according to Equation (2). □

Corollary 1. *Let $k, \alpha > 0$ be integers and let $s \in \mathbb{Z}_{2^\alpha}$ be odd. For a pair of distinct prime numbers p, q such that*

$$p \equiv q \equiv s \cdot (2^\alpha)^k + 1 \pmod{(2^\alpha)^{k+1}},$$

the system of congruences

$$\begin{aligned} x &\equiv (p-1)/2^\alpha \pmod{p-1}, \\ x &\equiv (q-1)/2^\alpha \pmod{q-1}, \end{aligned} \tag{3}$$

has solutions. Note that the solution is unique modulo $\operatorname{lcm}(p-1, q-1)$.

Proof. According to Theorem 1 the system of congruences (3) has solutions if and only if

$$(p-1)/2^{\alpha} \equiv (q-1)/2^{\alpha} \pmod{\gcd(p-1, q-1)}. \tag{4}$$

Equation (4) is equivalent to

$$\gcd(p-1, q-1) | (p-q)/2^{\alpha}.$$

and using Lemma 1 we obtain the desired result. □

Lemma 2. *Let $\alpha > 0$. We consider the set*

$$\mathcal{P}_i = \{p \text{ prime} \mid \exists k \in \mathbb{N} \text{ s.t. } p \equiv (2^i)^k + 1 \pmod{2^{i(k+1)}}\}.$$

Then there exists infinitely many primes $p \in \cap_{i=1}^{\alpha} \mathcal{P}_i$ and integers e, k_i such that

$$p \equiv 2^e + 1 \pmod{(2^i)^{k_i+1}},$$

for each $i \in [1, \alpha]$. More precisely, we have $e = \operatorname{lcm}(1, \ldots, \alpha)$ and $k_i = e/i$.

Proof. We begin by noticing that $\gcd(2^e+1, 2^{e+\alpha}) = 1$. According to Theorem 2, there exist infinitely many prime numbers p such that

$$p \equiv 2^e + 1 \pmod{2^{e+\alpha}}. \tag{5}$$

We can see that Equation (5) implies $p \equiv 2^e + 1 \pmod{2^{e+i}}$, for each $i \in [1, \alpha]$. This is due to the fact that $2^e + 1 < 2^{e+1} < 2^{e+2} < \ldots < 2^{e+\alpha}$.

If we can prove that $p \in \cap_{i=1}^{\alpha} \mathcal{P}_i$, then we can conclude our proof. Since $e = \operatorname{lcm}(1, 2, \ldots, \alpha)$, then there exist an integer k_i such that $e = k_i \cdot i$ for each $i \in [1, \alpha]$. As a result, we obtain that

$$p \equiv 2^e + 1 \pmod{(2^i)^{k_i+1}},$$

for each $i \in [1, \alpha]$. Therefore, $p \in \mathcal{P}_i$ for each $i \in [1, \alpha]$, which is equivalent to our conclusion. □

3.2 Description

Setup(λ): Set integers $k \geq 1$ and $e = \operatorname{lcm}(1, \ldots, k)$ such that $e + k < \lambda$. Randomly generate two distinct large prime numbers p, q such that $p, q \geq 2^{\lambda}$ and $p, q \equiv 2^e + 1 \pmod{2^{e+k}}$. Let $n = pq$. Select z_j, such that the following conditions hold

$$\begin{aligned} z_j &\equiv (p-1)/2^j \pmod{p-1}, \\ z_j &\equiv (q-1)/2^j \pmod{q-1}, \end{aligned} \tag{6}$$

where $j \in [1, k]$. Output the public key $pk = (n, y, k)$, where $y \in J_n \setminus QR_n$. The corresponding secret key is $sk = z$, where $z = \{z_j\}_{j \in [1,k]}$.

Encrypt(pk, m): To encrypt a message $m \in [0, 2^k)$, we choose $x \xleftarrow{\$} \mathbb{Z}_n^*$ and compute $c \equiv y^m x^{2^k} \bmod n$. Output the ciphertext c.
Decrypt(sk, c): To recover the message simply compute $m = Dec(z, y, c)$.

Algorithm 1: $Dec(z, y, c)$

Input: The secret value z, the value y and the ciphertext c
Output: The message m

$m \leftarrow 0$, $B \leftarrow 1$
foreach $j \in [1, k]$ **do**
  $v \leftarrow c^{z_j} \pmod n$
  $w \leftarrow (y^{z_j})^m \pmod n$
  if $v \neq w$ **then**
    $m \leftarrow m + B$
  end
  $B \leftarrow 2B$
end
return m

Correctness. Let $m = \sum_{w=0}^{k-1} b_w 2^w$ be the binary expansion of m. Note that

$$\begin{aligned} c^{z_j} &\equiv \left(\frac{c}{p}\right)_{2^j} = \left(\frac{y^m x^{2^k}}{p}\right)_{2^j} = \left(\frac{y^m}{p}\right)_{2^j} = \left(\frac{y}{p}\right)_{2^j}^{\sum_{w=0}^{j-1} b_w 2^w} \\ &\equiv (y^{z_j})^{\sum_{w=0}^{j-1} b_w 2^w} \pmod p \end{aligned}$$

since

1. $\left(\frac{x^{2^k}}{p}\right)_{2^j} = 1$, where $1 \leq j \leq k$;
2. $\sum_{w=0}^{k-1} b_w 2^w = \left(\sum_{w=0}^{j-1} b_w 2^w\right) + 2^j \cdot \left(\sum_{w=j}^{k-1} b_w 2^{w-j}\right)$.

Similarly, we obtain that

$$c^{z_j} \equiv (y^{z_j})^{\sum_{w=0}^{j-1} b_w 2^w} \pmod q.$$

Therefore, we obtain that

$$c^{z_j} \equiv (y^{z_j})^{\sum_{w=0}^{j-1} b_w 2^w} \pmod n.$$

As a result, the message m can be recovered bit by bit using z_j.

Remark 2. When $k = 1$ we obtain the Desmedt-Kurosawa encryption scheme [13].

Remark 3. Note that is sufficient to set the secret key only as $sk = z_k$, since the remaining values can be easily computed as $z_{k-j} = z_{k-j+1}^2$ for $j \in [1, k-1]$. But, for simplicity and clarity of the exposition, we describe it as such.

Remark 4. In the *Setup* phase, we have to select an y from $J_n \setminus QR_n$. An efficient way to perform this step is to randomly select $y_p \xleftarrow{\$} \mathbb{Z}_p^* \setminus QR_p$ and $y_q \xleftarrow{\$} \mathbb{Z}_q^* \setminus QR_q$, and then use the Chinese Remainder Theorem to compute the element $y \in \mathbb{Z}_n^*$ such that $y \equiv y_p \bmod p$ and $y \equiv y_q \bmod q$.

Optimized Decryption Algorithm. When studying Algorithm 1, we can observe that the values y^{z_j} are known beforehand. Hence, we can precompute $D_j = y^{z_j} \pmod n$ for $j \in [1, k]$ and augment the private key with these values.

3.3 Security Analysis

Theorem 3. *Assume that the* QR *and* SJS *assumptions hold. Then, the proposed scheme is* IND-CPA *secure in the standard model.*

Proof. To prove the statement, we simply change the distribution of the public key y. More precisely, instead of picking y uniformly from $J_n \setminus QR_n$, we choose it from the multiplicative subgroup of 2^k residues modulo n. According to the GR assumption, the adversary does not detect the difference between the original scheme and the one with the modified public key. In this case, the value c is not carrying any information about the message.

Formally, let A be an efficient PPT adversary, then there exist two efficient PPT algorithms B_1 and B_2 such that

$$ADV_A^{\text{IND-CPA}}(\lambda) \leq \frac{3}{2}\left((k - \frac{1}{3}) \cdot ADV_{B_1}^{\text{QR}}(\lambda) + (k-1) \cdot ADV_{B_2}^{\text{SJS}}(\lambda)\right).$$

Thus, the IND-CPA security of our proposed cryptosystem follows. □

Parameter Selection. In order for our scheme to work, we need to choose special primes $p, q \equiv 2^e + 1 \pmod{2^{e+k}}$. This means that the first least significant $e+k$ bits of both p and q are known to everybody. These facts have a very important impact in the security of the scheme. Due to a powerful attack described by Coppersmith [8] the size of $e + k$ must be at most $0.25 \log n$. Otherwise, it is possible to factor n.

3.4 Complexity Analysis

To facilitate our analysis, we consider that both primes have length λ when determining the ciphertext expansion and the encryption/decryption complexities. Considering the complexities listed in Table 1, our scheme achieves the performances presented in Table 2.

Keep in mind that, compared to the Desmedt-Kurasawa cryptosystem, our decryption algorithm makes one extra exponentiation when $k = 1$. Hence, this results in the gap between the complexity of the Desmedt-Kurasawa scheme and the complexity of our scheme when $k = 1$. Note that, compared to the analysis

Table 2. Performance analysis for an η-bit message

Scheme	Ciphertext size	Encryption Complexity
GM [21]	$2\lambda \cdot \eta$	$\mathcal{O}(2M(2\lambda) \cdot \eta)$
JL [23]	$2\lambda \cdot \left\lceil \frac{\eta}{k} \right\rceil$	$\mathcal{O}\left(2(k+1) \cdot M(2\lambda) \cdot \left\lceil \frac{\eta}{k} \right\rceil\right)$

Scheme	Decryption Complexity
GM [21]	$\mathcal{O}(\log(\lambda) \cdot M(\lambda) \cdot \eta)$
DK [13]	$\mathcal{O}(2\lambda \cdot M(2\lambda) \cdot \eta)$
JL [23]	$\mathcal{O}\left((2k\lambda + k) \cdot M(\lambda) \cdot \left\lceil \frac{\eta}{k} \right\rceil\right)$
This work	$\mathcal{O}\left((4k\lambda + k) \cdot M(2\lambda) \cdot \left\lceil \frac{\eta}{k} \right\rceil\right)$

provided in [28], we give a better decryption complexity for the Joye-Libert cryptosystem. More precicely,

$$\mathcal{O}\left((2k\lambda + k^2/2) \cdot M(\lambda) \cdot \left\lceil \frac{\eta}{k} \right\rceil\right) \text{ versus } \mathcal{O}\left((2k\lambda + k) \cdot M(\lambda) \cdot \left\lceil \frac{\eta}{k} \right\rceil\right).$$

The difference arises from considering only one exponentiation of y to the power $z_j m$ instead of two exponentiations, one to the power z_j and then one to the power m.

3.5 Implementation Details

We further provide the reader with benchmarks for our proposed PKE scheme. We ran each of the three sub-algorithms on a CPU Intel i7-4790 4.00 GHz and used GCC to compile it (with the O3 flag activated for optimization). Note that for all computations we used the GMP library [2] and the running times were calculated using the *omp_get_wtime()* function [1]. To obtain the average running time we chose to encrypt 100 128/192/256-bit messages, representing random symmetric keys. In order to have the same security as the symmetric keys we considered λ to be 1536/3840/15360, which according to NIST [3] offers a security strength of 128/192/256 bits.

According to our security analysis $e + k$ has to be less than 768/1920/3840. Using Lemma 2 we obtain that the first couples (k, e) are

$$(k, e) \in \{(1,1), (2,2), (3,6), (4,12), (5,60), (6,60), (7,420), \\ (8,840), (9,2520), (10,2520), (11,27720)\}.$$

Therefore, we have that k must be less than 8/9/11 when λ is 1536/3840/15360.

We further list our results in Tables 3 to 5 (run times are given in seconds). It should be noted that in Tables 3 to 5, the first lines of each algorithm correspond

to Algorithm 1, while the second ones to the optimized decryption version. When analyzing Table 3, note that in the case $k = 1$ we obtain the Desmedt-Kurosawa scheme.

For completeness, in Table 6 we also present the ciphertext size (in kilobytes $= 10^3$ bytes) for the previously mentioned parameters.

Table 3. Average running times (seconds) for a 128-bit message

Algorithm	$k = 1$	$k = 2$	$k = 4$
Setup	0.424845	0.471936	0.448168
	0.446417	0.468382	0.515123
Encrypt	0.006928	0.004558	0.003055
	0.007006	0.004507	0.003110
Decrypt	2.100240	2.606760	3.167010
	2.117050	2.113930	2.085170

Table 4. Average running times (seconds) for a 192-bit message

Algorithm	$k = 1$	$k = 2$	$k = 4$	$k = 8$
Setup	10.82220	12.45610	11.30620	10.73360
	10.13340	12.98770	11.90220	12.49030
Encrypt	0.041345	0.028024	0.020059	0.015736
	0.041092	0.028046	0.020045	0.015511
Decrypt	35.78030	44.64660	54.68320	54.66220
	35.58980	35.54510	35.44840	30.95290

Table 5. Average running times (seconds) for a 256-bit message

Algorithm	$k = 1$	$k = 2$	$k = 4$	$k = 8$
Setup	1259.440	1241.650	1381.090	1341.920
	1401.100	1191.060	1246.210	1475.490
Encrypt	0.461915	0.312051	0.223570	0.171349
	0.459074	0.310038	0.221316	0.169926
Decrypt	1520.860	1895.350	2308.220	2530.180
	1508.400	1499.500	1492.010	1435.410

Table 6. Ciphertext expansion

	$k=1$	$k=2$	$k=4$	$k=8$
$\lambda = 1536$	49.152	24.576	12.288	–
$\lambda = 3840$	184.320	92.160	46.080	23.040
$\lambda = 15360$	983.040	491.520	245.760	122.880

4 A Threshold Homomorphic Encryption Scheme

4.1 Description

For simplicity and clarity, we begin by describing a threshold protocol that requires a trusted dealer and is of type ℓ-out-of-ℓ. More precisely, we consider that the number of participants in our scheme is ℓ and that all of them are required to decrypt a ciphertext. On the other hand, if an adversary corrupts $\ell-1$ participants it is infeasible for him to decrypt a given ciphertext. The exact details of our protocol are provided below.

Dealing Phase: In the case of threshold decryption, the *Setup* phase of our PKE scheme is replaced by the following protocol.

1. First, the dealer sets integers $k \geq 1$ and $e = \text{lcm}(1, \ldots, k)$ such that $e+k < \lambda$. Then, he randomly generates two distinct large prime numbers p, q such that $p, q \geq 2^{\lambda}$ and $p, q \equiv 2^e + 1 \pmod{2^{e+k}}$. Finally, he sets $n = pq$.
2. Let $j \in [1, k]$. The dealer computes z_j, such that the system of congruences (6) holds. Then, he randomly chooses $z_{j,1}, z_{j,2}, \ldots, z_{j,\ell} \xleftarrow{\$} [0, 2^{2\lambda}]$ and computes $z_{j,0} = z_j - \sum_{i=1}^{\ell} z_{j,i}$. The public key of the protocol is $pk = (n, y, k, Z_0)$, where $y \in J_n \setminus QR_n$ and $Z_0 = \{z_{j,0}\}_{j\in[1,k]}$.
3. Lastly, the dealer sends the secret key share $Z_i = \{z_{j,i}\}_{j\in[1,k]}$ to player i for $i \in [1, \ell]$.

Decryption Phase: The decryption process of a ciphertext c proceeds as follows.

1. Player i computes $\beta_{j,i} \equiv c^{z_{j,i}} \pmod{n}$ for each $j \in [1, k]$ and broadcasts the vector $\beta_i = \{\beta_{j,i}\}_{j\in[1,k]}$.
2. All the players publicly compute the values $\beta_{j,0} = c^{z_{j,0}}$ for all $j \in [1, k]$.
3. Each player computes $C_j \equiv \prod_{i=0}^{\ell} \beta_{j,i} \pmod{n}$ and then it uses algorithm $Dec(z, y, c)$ to recover message m.

Correctness: In order to see why algorithm $Dec(z, y, c)$ works, all we have to prove is that $C_j \equiv c^{z_j} \pmod{n}$. Thus, we have

$$C_j \equiv \prod_{i=0}^{\ell} \beta_{j,i} \equiv \prod_{i=0}^{\ell} c^{z_{j,i}} \equiv c^{\sum_{i=0}^{\ell} z_{j,i}} \equiv c^{z_j} \pmod{n}.$$

Therefore, as is stated in Section 3.2, we are now able to decrypt the message bit by bit.

4.2 Security Analysis

Theorem 4. *The protocol presented in Section 4.1 is simulatable for any adversary who passively eavesdrops on at most $\ell - 1$ participant. Moreover, the protocol is* IND-CPA*, assuming the hardness of the* GR *assumption.*

Proof. To reduce the security of the threshold version of our PKE to the security of the original PKE, we need to show how an attacker for the original scheme $\mathcal{A}_o$ can simulate the view of $\ell - 1$ participants, and thus use the threshold attacker $\mathcal{A}_t$ to break the original scheme.

To achieve our goal, we need to show how $\mathcal{A}_o$ simulates the dealing phase. Below, we provide to the reader such a simulation. Furthermore, we assume that an adversary is able to corrupt $\ell - 1$ players and, without loss of generality, we assume that these players are $1, 2, \ldots, \ell - 1$.

Dealing Phase Simulation: When simulating the dealing phase, $\mathcal{A}_o$ takes as input an 2λ-bit integer n and an element $y \in J_n \setminus QR_n$, and it outputs the secret shares. Note that $\mathcal{A}_o$ is the dealer and it proceed as follows.

1. The dealer randomly chooses the integers $z_j^* \xleftarrow{\$} [0, 2^{2\lambda}]$, for $j \in [1, k]$.
2. Then, for each integer $j \in [1, k]$, the dealer randomly chooses the shares $z_{j,1}, z_{j,2}, \ldots, z_{j,\ell} \xleftarrow{\$} [0, 2^{2\lambda}]$ and computes $z_{j,0} = z_j^* - \sum_{i=1}^{\ell} z_{j,i}$. The public key of the protocol is $pk = (n, y, k, Z_0)$, where $Z_0 = \{z_{j,0}\}_{j \in [1,k]}$.
3. Lastly, the dealer sends the secret key share $Z_i = \{z_{j,i}\}_{j \in [1,k]}$ to player i for each $i \in [1, \ell)$.

To see why $\mathcal{A}_o$'s simulator does not decrease the probability of success of $\mathcal{A}_t$ we remark the following.

- For each $i \in [1, \ell)$, the elements $Z_i = \{z_{j,i}\}_{j \in [1,k]}$ have the same distribution as in the real execution of the protocol.
- The distribution of Z_0 is conditioned on the values Z_i seen by $\mathcal{A}_t$, where $i \in [1, \ell)$. Now, we notice that the distributions $z_j - z_{j,\ell}$ and $z_j^* - z_{j,\ell}$ are statistically indistinguishable for any $z_j, z_j^* \in [0, 2^{2\lambda}]$ when $z_{j,\ell}$ is randomly chosen from $[0, 2^{2\lambda}]$. Therefore, the distribution of Z_0 in the simulated execution of the protocol is statistically indistinguishable from the distribution of Z_0 in the real execution.

□

5 Extending the Threshold Encryption Scheme

5.1 Reducing the Threshold

We can construct an h-out-of-ℓ threshold scheme and achieve robustness[6] by using the same techniques shown in [26, Section 3.2]. For example, in the following we will show how to convert our ℓ-out-of-ℓ scheme into an h-out-of-ℓ threshold decryption scheme using the approach found in [36].

[6] *i.e.* even when a malicious adversary convinces $h - 1$ player to deviate arbitrarily from the protocol, the correct output can still be computed.

Dealing phase:

1. The dealer generates parameters p, q, n, y and $\{z_{j,i}\}_{j\in[1,k],i\in[0,\ell]}$ as in Section 4.1.
2. The dealer chooses a prime $P > n$ and broadcasts the public key $pk = (n, y, k, Z_0, P)$.
3. Before computing the secret shares, for each $z_{j,i}$, he randomly generates an $(h-1)$-degree polynomial $f_{j,i}$ over the field $\mathbb{Z}_P$ such that $f_{j,i}(0) = z_{j,i}$.
4. Lastly, the dealer sends the secret shares Z_i and $F_i = \{f_{j,i}(i)\}_{j\in[1,k]}$ to player i for $i \in [1, \ell]$.

Decryption Phase:

1. The decryption process proceeds as in Section 4.1.
2. If player i fails to participate, then the remaining players can publicly reconstruct Z_i using the shares they have been given. Therefore, using interpolation the vector β_i can be computed publicly and included in the computation of C_j.

The technique presented in [36] is a generic approach to convert an ℓ-out-of-ℓ scheme into an h-out-of-ℓ one. Therefore, we obtain the following result.

Theorem 5. *The h-out-of-ℓ extension presented in Section 5.1 is simulatable for any adversary who passively eavesdrops on at most $h-1$ participant. Moreover, the protocol is* IND-CPA*, assuming the hardness of the* GR *assumption.*

5.2 Chosen-Ciphertext Security

Using the generic construction presented in [16], our threshold cryptosystem[7] can be converted from a scheme secure against chosen plaintext attacks into one secure against chosen ciphertext attacks. The twin-encryption technique requires two independent runs of the initial threshold PKE and a non-interactive zero-knowledge proof that two ciphertexts encrypt the same plaintext. We further provide such a NIZK.

For $i \in [0, 1]$, let $a_i \equiv y_i^m x_i^{2^k} \pmod{n_i}$ be two ciphertexts encrypting the same message m using the public keys $pk_i = (n_i, y_i, k)$[8]. Also, let H be a cryptographic hash function which outputs values in the range $[0, B]$. Classical cryptographic hash function have $B = 2^{256}/2^{384}/2^{512}$. Then the NIZK protocol works as follows. Note that in the protocol parameter A must selected such that it is significantly larger than $2^k B$ since it defines the size of some random data used to mask the secret.

Generate Proof (pk_0, pk_1, a_0, a_1): Let $i \in [0, 1]$. To generate a proof, we first randomly select $r \xleftarrow{\$} [0, A]$ and $\chi_i \xleftarrow{\$} \mathbb{Z}_{n_i}^*$. Then we compute $\alpha_i \leftarrow y_i^r \chi_i^{2^k} \pmod{n_i}$ and $\beta \leftarrow H(pk_0, pk_1, a_0, a_1, \alpha_0, \alpha_1)$. Finally, we compute $s \leftarrow r + m\beta$ and $u_i \leftarrow \chi_i x_i^\beta \pmod{n_i}$. A proof of equality is the tuple $S = (\beta, s, u_0, u_1)$.

[7] also, the Joye-Libert PKE

[8] Z_0 is not used for encryption, and therefore we omit it for simplicity.

Verify Proof$(pk_0, pk_1, a_0, a_1, S)$: Before verifying the proof, we must first validate S. More precisely, we must check that $\beta \in [0, B]$, $u_i \in \mathbb{Z}_{n_i}$ and $s \in [0, A+(2^k-1)B]$. Then, to verify the proof S we simply have to compute $v_i \leftarrow y_i^s u_i^{2^k} / a_i^\beta \pmod{n_i}$ and $\gamma \leftarrow H(pk_0, pk_1, a_0, a_1, v_0, v_1)$, and check if γ is equal to β.

Completeness. If S is a valid proof, then the following relations are true

$$v_i \equiv \frac{y_i^s u_i^{2^k}}{a_i^\beta} \equiv \frac{y_i^{r+m\beta}(\chi_i x_i^\beta)^{2^k}}{(y_i^m x_i^{2^k})^\beta} \equiv y_i^r \chi_i^{2^k} \equiv \alpha_i \pmod{n_i},$$

and thus γ must be equal to β.

Before proving that the NIZK presented in Section 5 is simulation-sound, we first need to adapt the forking lemma presented in [34,35] to our setting.

Lemma 3. *Let A be a PPT algorithm, given only the public data as input and which can ask q_h queries to the random oracle. If A can find in time t a valid proof $(pk_0, pk_1, a_0, a_1, \beta, s, u_0, u_1)$ for an invalid word (a_0, a_1) with probability $\nu \geq 7q_h/B$, then within time $t' \leq 16q_h t/\nu$ and with probability $\nu' \geq 1/9$, a replay of A outputs a distinct second proof $(pk_0, pk_1, a_0, a_1, \bar{\beta}, \bar{s}, \bar{u}_0, \bar{u}_1)$ for the invalid word (a_0, a_1) such that $\beta \neq \bar{\beta}$.*

Theorem 6. *The NIZK presented in Section 5.2 is simulation-sound in the random oracle.*

Proof. The first step we need to take is to create a list of accepted proofs. In order to do this, we randomly select $\beta \xleftarrow{\$} [0, B]$, $s \xleftarrow{\$} [0, A+(2^k-1)B]$, $u_i \xleftarrow{\$} \mathbb{Z}_{n_i}$ and we define

$$H(pk_0, pk_1, a_0, a_1, y_0^s u_0^{2^k} / a_0^\beta \pmod{n_0}, y_1^s u_1^{2^k} / a_1^\beta \pmod{n_1}) \leftarrow \beta.$$

This trick is possible due to working in the random oracle model.

Now, let assume that a PPT adversary A is able to forge a new proof for a wrong word (a_0, a_1) within the time bound t and with probability ν. Using Lemma 3 we obtain two such forgeries

$$(pk_0, pk_1, a_0, a_1, \beta, s, u_0, u_1) \text{ and } (pk_0, pk_1, a_0, a_1, \bar{\beta}, \bar{s}, \bar{u}_0, \bar{u}_1).$$

Assuming that A has not broken the collision intractability of H, we obtain

$$\frac{y_i^s u_i^{2^k}}{a_i^\beta} \equiv \frac{y_i^{\bar{s}} \bar{u}_i^{2^k}}{a_i^{\bar{\beta}}} \pmod{n_i}.$$

which is equivalent with

$$y_i^{\tilde{s}} \tilde{u}_i^{2^k} \equiv a_i^{\tilde{\beta}} \pmod{n_i}, \tag{7}$$

where $\tilde{s} = s - \bar{s}$, $\tilde{u}_i = u_i \bar{u}_i^{-1}$ and $\tilde{\beta} = \beta - \bar{\beta}$.

Since (a_0, a_1) is an invalid word, then $a_i \equiv y_i^{m_i} x_i^{2^k} \pmod{n_i}$, where $m_0 \neq m_1$. Rewriting Equation (7), we obtain

$$y_i^{\tilde{s}} \tilde{u}_i^{2^k} \equiv (y_i^{m_i} x_i^{2^k})^{\tilde{\beta}} \pmod{n_i},$$

which is equivalent with

$$y_i^{\tilde{s}-m_i\tilde{\beta}} \equiv (\tilde{u}_i^{-1} x_i^{\tilde{\beta}})^{2^k} \pmod{n_i}.$$

Since $y_i \in J_{n_i} \setminus QR_{n_i}$, we obtain that

$$\tilde{s} - m_i\tilde{\beta} \equiv 0 \pmod{2^k},$$

and thus $m_0 \equiv m_1 \pmod{2^k}$. Therefore, the word is in the language, unless one has broken the collision intractability for H. Using the random oracle assumption and the birthday paradox, we obtain that in order to get a probability of obtaining a collision greater than $1/9$, the adversary has to ask more than $\sqrt{B}/3$ queries to H. Hence,

$$\frac{\sqrt{B}}{3}\tau \leq t' \leq \frac{16 q_h t}{\nu}, \tag{8}$$

where τ is the time required for an evaluation of H. Rewriting Equation (8), we obtain that

$$ADV_A^{\text{SIM-NIZK}}(\lambda) \leq \nu \leq \frac{48 q_h t}{\tau\sqrt{B}}.$$

Therefore, our non-interactive proof is simulation sound. □

6 Conclusions

In this paper we have constructed a novel variant of the Joye-Libert cryptosystem that allows an user to decrypt messages even if he does not know the factorization of the composite modulus. Based on this variant, we showed how to achieve threshold decryption for the Joye-Libert cryptosystem, and therefore solving some open problems stated in [17, 23, 26].

In the second part of the paper, we present several extensions of our basic threshold scheme. We first provide an example of converting the ℓ-out-of-ℓ threshold into an h-out-of-ℓ one. Then, we provide a non-interactive zero-knowledge protocol that can be used to protect the proposed cryptosystems from chosen ciphertext attacks. Note that our NIZK can also be used to protect the Desmedt-Kurasawa PKE, and thus filling a gap left by the authors in [13].

Future Work. A possible method for accelerating our proposed systems would be to use small multiple primes instead of only two primes. Therefore, an interesting research direction would be to find a method to modify the multi-prime Joye-Libert version proposed in [28,39] such that it allows decryption without knowing the factorization of n.

References

1. OpenMP. https://www.openmp.org/
2. The GNU Multiple Precision Arithmetic Library. https://gmplib.org/
3. Barker, E.: NIST SP800-57 Recommendation for Key Management, Part 1: General. Tech. rep., NIST (2016)
4. Benhamouda, F., Herranz, J., Joye, M., Libert, B.: Efficient Cryptosystems from 2^k-th Power Residue Symbols. Journal of Cryptology **30**(2), 519–549 (2017)
5. Boneh, D., Franklin, M.: Efficient Generation of Shared RSA Keys (Extended Abstract). In: CRYPTO 1997. Lecture Notes in Computer Science, vol. 1294, pp. 425–439. Springer (1997)
6. Canetti, R., Goldwasser, S.: An Efficient Threshold Public Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack. In: EUROCRYPT 1999. Lecture Notes in Computer Science, vol. 1592, pp. 90–106. Springer (1999)
7. Cohen, J., Fischer, M.: A Robust and Verifiable Cryptographically Secure Ellection Scheme (extended abstract). In: FOCS 1985. pp. 372–382. IEEE Computer Society Press (1985)
8. Coppersmith, D.: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. Journal of Cryptology **10**(4), 233–260 (1997)
9. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty Computation from Threshold Homomorphic Encryption. In: EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 280–300. Springer (2001)
10. Crandall, R., Pomerance, C.: Prime Numbers: A Computational Perspective. Number Theory and Discrete Mathematics, Springer (2005)
11. Damgård, I., Jurik, M.: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In: PKC 2001. Lecture Notes in Computer Science, vol. 1992, pp. 119–136. Springer (2001)
12. Desmedt, Y., Frankel, Y.: Threshold Cryptosystems. In: CRYPTO 1989. Lecture Notes in Computer Science, vol. 435, pp. 307–315. Springer (1989)
13. Desmedt, Y., Kurosawa, K.: A Generalization and a Variant of Two Threshold Cryptosystems Based on Factoring. In: ISC 2007. Lecture Notes in Computer Science, vol. 4779, pp. 351–361. Springer (2007)
14. ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
15. Feige, U., Lapidot, D., Shamir, A.: Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract). In: FOCS 1990. pp. 308–317. IEEE Computer Society (1990)
16. Fouque, P.A., Pointcheval, D.: Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In: ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 351–368. Springer (2001)
17. Fouque, P.A., Poupard, G., Stern, J.: Sharing Decryption in the Context of Voting or Lotteries. In: Financial Cryptography. Lecture Notes in Computer Science, vol. 1962, pp. 90–104. Springer (2000)
18. Franklin, M.K., Haber, S.: Joint Encryption and Message-Efficient Secure Computation. In: CRYPTO 1993. Lecture Notes in Computer Science, vol. 773, pp. 266–277. Springer (1993)
19. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust Threshold DSS Signatures. In: EUROCRYPT 1996. Lecture Notes in Computer Science, vol. 1070, pp. 354–371. Springer (1996)

20. Girault, M., Poupard, G., Stern, J.: On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. Journal of Cryptology **19**(4), 463–487 (2006)
21. Goldwasser, S., Micali, S.: Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In: STOC 1982. pp. 365–377. ACM (1982)
22. Goldwasser, S., Micali, S.: Probabilistic Encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984)
23. Joye, M., Libert, B.: Efficient Cryptosystems from 2^k-th Power Residue Symbols. In: EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 76–92. Springer (2013)
24. Joye, M., Libert, B.: Efficient Cryptosystems from 2^k-th Power Residue Symbols. IACR Cryptology ePrint Archive **2013/435** (2014)
25. Katz, J., Myers, S., Ostrovsky, R.: Cryptographic Counters and Applications to Electronic Voting. In: EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 78–92. Springer (2001)
26. Katz, J., Yung, M.: Threshold Cryptosystems Based on Factoring. In: ASIACRYPT 2002. Lecture Notes in Computer Science, vol. 2501, pp. 192–205. Springer (2002)
27. Kennard, L.: Two Classic Theorems from Number Theory: The Prime Number Theorem and Dirichlet's Theorem (2006)
28. Maimuţ, D., Teşeleanu, G.: A New Generalisation of the Goldwasser-Micali Cryptosystem Based on the Gap 2^{k}-Residuosity Assumption. In: SecITC 2020. Lecture Notes in Computer Science, vol. 12596, pp. 24–40. Springer (2020)
29. Naccache, D., Stern, J.: A New Public Key Cryptosytem Based on Higher Residues. In: CCS 1998. pp. 59–66. ACM (1998)
30. Naor, M., Yung, M.: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: STOC 1990. pp. 427–437. ACM (1990)
31. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: EUROCRYPT 1999. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer (1999)
32. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party. In: EUROCRYPT 1991. Lecture Notes in Computer Science, vol. 547, pp. 522–526. Springer (1991)
33. Pei, D., Salomaa, A., Ding, C.: Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography. World Scientific Publishing (1996)
34. Pointcheval, D., Stern, J.: Security Proofs for Signature Schemes. In: EUROCRYPT 1996. Lecture Notes in Computer Science, vol. 1070, pp. 387–398. Springer (1996)
35. Pointcheval, D., Stern, J.: Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology **13**(3), 361–396 (2000)
36. Rabin, T.: A Simplified Approach to Threshold and Proactive RSA. In: CRYPTO 1998. Lecture Notes in Computer Science, vol. 1462, pp. 89–104. Springer (1998)
37. Sahai, A.: Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In: FOCS 1999. pp. 543–553. IEEE Computer Society (1999)
38. Shoup, V., Gennaro, R.: Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In: EUROCRYPT 1998. Lecture Notes in Computer Science, vol. 1403, pp. 1–16. Springer (1998)
39. Teşeleanu, G.: The Case of Small Prime Numbers Versus the Joye-Libert Cryptosystem. Mathematics **10**(9) (2022)
40. Yan, S.Y.: Number Theory for Computing. Theoretical Computer Science, Springer (2002)