

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

George Teşeleanu<sup>1,2</sup> 

<sup>1</sup>Advanced Technologies Institute, 10 Dinu Vintilă, Bucharest, Romania

<sup>2</sup>Simion Stoilow Institute of Mathematics of the Romanian Academy, 21 Calea Grivitei, Bucharest, Romania

Keywords: Threshold Decryption, Homomorphic Encryption, Gap Residuosity Assumption.

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 (Boneh and Franklin, 1997; Fouque et al., 2000; Fouque and Pointcheval, 2001; Damgård and Jurik, 2001) and discrete logarithm-based schemes (Desmedt and Frankel, 1989; Fouque and Pointcheval, 2001; Pedersen, 1991; Shoup and Gennaro, 1998; Canetti and Goldwasser, 1999). But according to (Katz and Yung, 2002; Fouque et al., 2000; Cramer et al., 2001), there is still a need to design threshold schemes for many specific cryptosystems. Furthermore, as many have pointed out previously (Franklin and Haber, 1993; Fouque et al., 2000; Damgård and Jurik, 2001; Katz et al., 2001; Cramer et al., 2001), 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 (Katz and Yung, 2002)

developed a threshold cryptosystem based on the Goldwasser-Micali encryption scheme (Goldwasser and Micali, 1982; Goldwasser and Micali, 1984). Moreover, their conversion keeps the homomorphic properties of the original scheme. The Katz-Yung scheme is revisited in (Desmedt and Kurosawa, 2007) 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 (Joye and Libert, 2013) and it was reconsidered in (Benhamouda et al., 2017). Despite the fact that it is simple and elegant, the Goldwasser-Micali scheme is quite uneconomical in terms of bandwidth<sup>1</sup>. 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 (Naccache and Stern, 1998; Cohen and Fischer, 1985) and efficiently supports the encryption of larger messages. The authors of (Joye and Libert, 2013) leave as an open problem the extension of their scheme, starting from (Katz and Yung, 2002), 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 (Joye and Libert, 2013; Benhamouda et al., 2017) that generalizes both the Katz-Yung scheme (Katz and Yung, 2002) and the Desmedt-Kurasawa scheme (Desmedt

<sup>1</sup> $k \cdot \log_2 n$  bits are needed to encrypt a  $k$ -bit message, where  $n$  is a composite modulus as described in (Goldwasser and Micali, 1982; Goldwasser and Micali, 1984)

 <https://orcid.org/0000-0003-3953-2744>

and Kurosawa, 2007). 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 (see the full paper). This topic was tackled by Katz and Yung for their scheme (Katz and Yung, 2002). More precisely, they applied the generic conversion method from (Fouque and Pointcheval, 2001) 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 Kurosawa (Desmedt and Kurosawa, 2007) 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 (Fouque and Pointcheval, 2001) and the signature protocol from (Girault et al., 2006), 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-Kurosawa schemes.

**Full Version.** The full version of the paper can be found here (Teşeleanu, 2023).

**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. We conclude in Section 5.

## 2 PRELIMINARIES

**Notations.** Throughout the paper,  $\lambda$  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  $J(a, n)$ .  $J_n$  and  $\bar{J}_n$  de-

note the sets of integers modulo  $n$  with Jacobi 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  $\mathbb{Z}_p = \{-(p-1)/2, \dots, -1, 0, 1, \dots, (p-1)/2\}$ . The set of integers  $\{0, \dots, a-1\}$  is further denoted by  $[0, a)$ . For shorthand, we denote the set  $[0, a+1)$  by  $[0, a]$ . Multidimensional vectors  $v = (v_0, \dots, v_{s-1})$  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 (Yan, 2002). The classical Legendre symbol is obtained when  $k = 1$ .

**Definition 2.1.** Let  $p$  be an odd prime such that  $2^k | p-1$ . Then the symbol

$$J_{2^k}(a, p) = a^{\frac{p-1}{2^k}} \pmod{p}$$

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

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

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

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 (Pei et al., 1996).

**Theorem 2.1 (Generalized Chinese Remainder Theorem).** Let  $m_1, m_2, \dots, m_t$  be positive integers. For a set of integers  $a_1, a_2, \dots, a_t$  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  $\text{lcm}(m_1, m_2, \dots, m_t)$ .

We additionally use a theorem proved by Dirichlet in 1837. This theorem establishes the constraints necessary for the existence of infinitely many primes in an arithmetic progression. The original proof can be found in (Kennard, 2006).

**Theorem 2.2** (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 (Crandall and Pomerance, 2005). 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  $\mu$ -bit numbers and  $k$ -bit exponents.

Operation	Complexity
Multiplication	$O(\mu \log(\mu) \log(\log(\mu)))$
Exponentiation	$O(kM(\mu))$
Jacobi symbol	$O(\log(\mu)M(\mu))$

## 2.3 Security Assumptions

**Definition 2.2** (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  $(x, k, n)$  if  $x \in J_n \setminus QR_n$ . We define the advantage

$$ADV_{A,k}^{\text{GR}}(\lambda) = \left| \Pr[A(x, k, n) = 1 \mid x \xleftarrow{\$} J_n \setminus QR_n] - \Pr[A(x^k, k, n) = 1 \mid x \xleftarrow{\$} \mathbb{Z}_n^*] \right|.$$

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

## 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 2.3** (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**( $\lambda$ ): 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

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

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  $ADV_A^{\text{IND-CPA}}(\lambda)$  is negligible.

**Definition 2.4** (Indistinguishability under Chosen Ciphertext Attacks - IND-CCA). In the context of Definition 2.3, 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  $c$ .

### 2.4.1 The Joye-Libert PKE Scheme

The Joye-Libert scheme was introduced in (Joye and Libert, 2013) as a generalization of the Goldwasser-Micali cryptosystem (Goldwasser and Micali, 1982) to multi-bit messages. The scheme is proven secure in the standard model under the GR assumption (Joye and Libert, 2013; Benhamouda et al., 2017). We shortly describe the algorithms of the Joye-Libert cryptosystem.

**Setup**( $\lambda$ ): Set an integer  $k \geq 1$ . Randomly generate two distinct large prime numbers  $p, q$  such that  $p, q \geq 2^\lambda$  and  $p, q \equiv 1 \pmod{2^k}$ . 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  $c \equiv y^m x^{2^k} \pmod{n}$ . Output the ciphertext  $c$ .

**Decrypt**( $sk, c$ ): Compute  $z \equiv J_{2^k}(c, p)$  and find  $m$  such that the relation  $[J_{2^k}(y, p)]^m \equiv z \pmod{p}$  holds. Efficient methods to recover  $m$  can be found in (Joye and Libert, 2014).

### 2.4.2 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  $\ell$  and the decryption threshold  $h$ ; it outputs the public key and distributes the shares of the secret key to the  $\ell$  players. The *Decrypt* algorithm takes as input a ciphertext; it forwards it to player  $i$ 's decryption algorithm<sup>2</sup>; 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 (Gennaro *et al.*, 1996). 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.

## 3 A PUBLIC KEY ENCRYPTION SCHEME

### 3.1 Prerequisites

*Lemma 3.1.* 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

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

<sup>2</sup>which has access to player  $i$ 's secret key share

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

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

Since  $s$  is odd, we have that

$$2^{\alpha k} | p - 1, 2^{\alpha k+1} \nmid p - 1 \text{ and } 2^{\alpha k} | q - 1, 2^{\alpha k+1} \nmid q - 1,$$

and thus

$$2^{\alpha k} | \gcd(p - 1, q - 1) \text{ and } 2^{\alpha k+1} \nmid \gcd(p - 1, q - 1).$$

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).  $\square$

*Corollary 3.1.1.* 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

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

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

*Proof.* According to Theorem 2.1 the system of congruences (3) has solutions if and only if

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

Equation (4) is equivalent to

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

and using Lemma 3.1 we obtain the desired result.  $\square$

*Lemma 3.2.* 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  $e = \text{lcm}(1, \dots, \alpha)$  and  $k_i = e/i$ .

*Proof.* We begin by noticing that  $\gcd(2^e + 1, 2^{e+\alpha}) = 1$ . According to Theorem 2.2, there exist infinitely many prime numbers  $p$  such that

$$p \equiv 2^e + 1 \pmod{2^{e+\alpha}}. \quad (5)$$

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  $2^e + 1 < 2^{e+1} < 2^{e+2} < \dots < 2^{e+\alpha}$ .

If we can prove that  $p \in \cap_{i=1}^{\alpha} \mathcal{P}_i$ , then we can conclude our proof. Since  $e = \text{lcm}(1, 2, \dots, \alpha)$ , 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.  $\square$

### 3.2 Description

*Setup*( $\lambda$ ): Set integers  $k \geq 1$  and  $e = \text{lcm}(1, \dots, k)$  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

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

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  $c \equiv y^m x^{2^k} \pmod{n}$ . Output the ciphertext  $c$ .

*Decrypt*( $sk, c$ ): To recover the message simply compute  $m = \text{Dec}(z, y, c)$ .

---

Algorithm 1:  $\text{Dec}(Z, y, c)$ .

---

**Input:** The secret value  $z$  and the ciphertext  $c$

**Output:** The message  $m$

```

1  $m \leftarrow 0, B \leftarrow 1$ 
2 foreach  $j \in [1, k]$  do
3    $v \leftarrow c^{z_j} \pmod{n}$ 
4    $w \leftarrow (y^{z_j})^m \pmod{n}$ 
5   if  $v \neq w$  then
6      $m \leftarrow m + B$ 
7    $B \leftarrow 2B$ 
8 return  $m$ 
    
```

---

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

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

since

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

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.* When  $k = 1$  we obtain the Desmedt-Kurosawa encryption scheme (Desmedt and Kurosawa, 2007).

*Remark.* 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.* 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  $y \equiv y_p \pmod{p}$  and  $y \equiv y_q \pmod{q}$ .

**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 3.3.* 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

$$\begin{aligned} \text{ADV}_A^{\text{IND-CPA}}(\lambda) &\leq \frac{3}{2} \left( (k-1) \text{ADV}_{B_1}^{\text{QR}}(\lambda) \right. \\ &\quad \left. + (k-1) \text{ADV}_{B_2}^{\text{SJS}}(\lambda) \right). \end{aligned}$$

Thus, the IND-CPA security of our proposed cryptosystem follows.  $\square$



**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 (Coppersmith, 1997) 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  $\lambda$  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. Note that GM, JL and DK are presented in (Goldwasser and Micali, 1982), (Joye and Libert, 2013) and (Desmedt and Kurosawa, 2007), respectively.

Table 2: Performance analysis for an  $\eta$ -bit message.

Scheme	Ciphertext size	Encryption Complexity
GM	$2\lambda\eta$	$O(2M(2\lambda)\eta)$
JL	$2\lambda \lceil \frac{\eta}{k} \rceil$	$O\left(2(k+1)M(2\lambda) \lceil \frac{\eta}{k} \rceil\right)$
Scheme	Decryption Complexity	
GM	$O(\log(\lambda)M(\lambda)\eta)$	
DK	$O(2\lambda M(2\lambda)\eta)$	
JL	$O\left((2k\lambda + k)M(\lambda) \lceil \frac{\eta}{k} \rceil\right)$	
<b>This work</b>	$O\left((4k\lambda + k)M(2\lambda) \lceil \frac{\eta}{k} \rceil\right)$	

### 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 (gmp, ) and the running times were calculated using the `omp_get_wtime()` function (omp, ). 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  $\lambda$  to be 1536/3840/15360, which according

to NIST (Barker, 2016) 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 3.2 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  $\lambda$  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.

	$k = 1$	$k = 2$	$k = 4$
Setup	0.42484	0.47193	0.44816
	0.44641	0.46838	0.51512
Encrypt	0.00692	0.00455	0.00305
	0.00700	0.00450	0.00311
Decrypt	2.10024	2.60676	3.16701
	2.11705	2.11393	2.08517

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

	$k = 1$	$k = 2$	$k = 4$	$k = 8$
Setup	10.8222	12.4561	11.3062	10.7336
	10.1334	12.9877	11.9022	12.4903
Encrypt	0.04134	0.02802	0.02005	0.01573
	0.04109	0.02804	0.02004	0.01551
Decrypt	35.7803	44.6466	54.6832	54.6622
	35.5898	35.5451	35.4484	30.9529

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

	$k = 1$	$k = 2$	$k = 4$	$k = 8$
Setup	1259.44	1241.65	1381.09	1341.92
	1401.10	1191.06	1246.21	1475.49
Encrypt	0.46191	0.31205	0.22357	0.17134
	0.45907	0.31003	0.22131	0.16992
Decrypt	1520.86	1895.35	2308.22	2530.18
	1508.40	1499.50	1492.01	1435.41

Table 6: Ciphertext expansion.

	$k = 1$	$k = 2$	$k = 4$	$k = 8$
$\lambda = 1536$	49.15	24.57	12.28	—
$\lambda = 3840$	184.32	92.16	46.08	23.04
$\lambda = 15360$	983.04	491.52	245.76	122.88

## 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  $\ell$ -out-of- $\ell$ . More precisely, we consider that the number of participants in our scheme is  $\ell$  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  $e = \text{lcm}(1, \dots, k)$  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  $z_{j,1}, z_{j,2}, \dots, z_{j,\ell} \xleftarrow{\$} [0, 2^{2\lambda}]$  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

The proof of our result can be found in (Teşeleanu, 2023).

*Theorem 4.1.* 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.

## 5 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 (Joye and Libert, 2013; Fouque et al., 2000; Katz and Yung, 2002).

In the full paper, we also present several extensions of our basic threshold scheme. We first provide an example of converting the  $\ell$ -out-of- $\ell$  threshold into an  $h$ -out-of- $\ell$  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 (Desmedt and Kurosawa, 2007).

**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 (Maimuţ and Teşeleanu, 2020; Teşeleanu, 2022) such that it allows decryption without knowing the factorization of  $n$ .

## REFERENCES

OpenMP. <https://www.openmp.org/>.

- The GNU Multiple Precision Arithmetic Library. <https://gmplib.org/>.
- Barker, E. (2016). NIST SP800-57 Recommendation for Key Management, Part 1: General. Technical report, NIST.
- Benhamouda, F., Herranz, J., Joye, M., and Libert, B. (2017). Efficient Cryptosystems from  $2^k$ -th Power Residue Symbols. *Journal of Cryptology*, 30(2):519–549.
- Boneh, D. and Franklin, M. (1997). Efficient Generation of Shared RSA Keys (Extended Abstract). In *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 425–439. Springer.
- Canetti, R. and Goldwasser, S. (1999). An Efficient Threshold Public Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack. In *EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 90–106. Springer.
- Cohen, J. and Fischer, M. (1985). A Robust and Verifiable Cryptographically Secure Election Scheme (extended abstract). In *FOCS 1985*, pages 372–382. IEEE Computer Society Press.
- Coppersmith, D. (1997). Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology*, 10(4):233–260.
- Cramer, R., Damgård, I., and Nielsen, J. B. (2001). Multiparty Computation from Threshold Homomorphic Encryption. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–300. Springer.
- Crandall, R. and Pomerance, C. (2005). *Prime Numbers: A Computational Perspective*. Number Theory and Discrete Mathematics. Springer.
- Damgård, I. and Jurik, M. (2001). A Generalisation, a Simplification and Some Applications of Paillier’s Probabilistic Public-Key System. In *PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer.
- Desmedt, Y. and Frankel, Y. (1989). Threshold Cryptosystems. In *CRYPTO 1989*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer.
- Desmedt, Y. and Kurosawa, K. (2007). A Generalization and a Variant of Two Threshold Cryptosystems Based on Factoring. In *ISC 2007*, volume 4779 of *Lecture Notes in Computer Science*, pages 351–361. Springer.
- Fouque, P.-A. and Pointcheval, D. (2001). Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 351–368. Springer.
- Fouque, P.-A., Poupard, G., and Stern, J. (2000). Sharing Decryption in the Context of Voting or Lotteries. In *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 90–104. Springer.
- Franklin, M. K. and Haber, S. (1993). Joint Encryption and Message-Efficient Secure Computation. In *CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 266–277. Springer.
- Gennaro, R., Jarecki, S., Krawczyk, H., and Rabin, T. (1996). Robust Threshold DSS Signatures. In *EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 354–371. Springer.
- Girault, M., Poupard, G., and Stern, J. (2006). On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. *Journal of Cryptology*, 19(4):463–487.
- Goldwasser, S. and Micali, S. (1982). Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *STOC 1982*, pages 365–377. ACM.
- Goldwasser, S. and Micali, S. (1984). Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299.
- Joye, M. and Libert, B. (2013). Efficient Cryptosystems from  $2^k$ -th Power Residue Symbols. In *EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 76–92. Springer.
- Joye, M. and Libert, B. (2014). Efficient Cryptosystems from  $2^k$ -th Power Residue Symbols. *IACR Cryptology ePrint Archive*, 2013/435.
- Katz, J., Myers, S., and Ostrovsky, R. (2001). Cryptographic Counters and Applications to Electronic Voting. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 78–92. Springer.
- Katz, J. and Yung, M. (2002). Threshold Cryptosystems Based on Factoring. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 192–205. Springer.
- Kennard, L. (2006). *Two Classic Theorems from Number Theory: The Prime Number Theorem and Dirichlet’s Theorem*.
- Maimuț, D. and Teșeleanu, G. (2020). A New Generalisation of the Goldwasser-Micali Cryptosystem Based on the Gap  $2^k$ -Residuosity Assumption. In *SecITC 2020*, volume 12596 of *Lecture Notes in Computer Science*, pages 24–40. Springer.
- Naccache, D. and Stern, J. (1998). A New Public Key Cryptosystem Based on Higher Residues. In *CCS 1998*, pages 59–66. ACM.
- Pedersen, T. P. (1991). A Threshold Cryptosystem without a Trusted Party. In *EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 522–526. Springer.
- Pei, D., Salomaa, A., and Ding, C. (1996). *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*. World Scientific Publishing.
- Shoup, V. and Gennaro, R. (1998). Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 1–16. Springer.
- Teșeleanu, G. (2022). The Case of Small Prime Numbers Versus the Joye-Libert Cryptosystem. *Mathematics*, 10(9).
- Teșeleanu, G. (2023). Threshold Cryptosystems Based on  $2^k$ -th Power Residue Symbols. *IACR Cryptology ePrint Archive*, 2023/601.
- Yan, S. Y. (2002). *Number Theory for Computing*. Theoretical Computer Science. Springer.