One-Way Homomorphic Encryption: A Composite Group Approach

Mahdi Mahdavi[1,2] and Helena Rifà-Pous[1]

[1] Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Spain
[2] Shahid Beheshti University (SBU), Iran
{m_mahdavi,hrifa}@uoc.edu

Abstract. Homomorphic Encryption (HE) is a fundamental Privacy-Enhancing Technology (PET) that enables computations on encrypted data without decryption. Despite its utility, designing an efficient and secure HE scheme is highly complex, requiring sophisticated cryptographic techniques. This paper introduces a novel approach to achieving homomorphic properties—supporting either one addition or one multiplication—within composite groups. While the proposed technique exhibits one-wayness, it has a good potential to serve as a foundational building block for constructing indistinguishable cryptosystems. This work contributes to the advancement of homomorphic encryption by providing a new perspective on secure computation within structured algebraic settings.

Keywords: Homomorphic Encryption · One way function · Composite group · Rabin Cryptosystem · Quadratic residue.

1 Introduction

Fully Homomorphic Encryption (FHE) is a fundamental cryptographic tool that ensures data confidentiality during computation, allowing operations on encrypted data without requiring decryption. Several FHE schemes have been developed based on different hardness assumptions, including Ideal Lattices [8,16], Learning with Errors (LWE) [3,2,7,4,5], and Approximate Greatest Common Divisor (AGCD) [6,13].

In contrast, homomorphic encryption schemes within composite groups generally provide only partial homomorphism, with the exception of the Boneh-Goh-Nissim (BGN) scheme [1]. The BGN scheme supports the evaluation of any 2-DNF (Disjunctive Normal Form) function, but it relies on pairings over composite-order groups, which involve computationally intensive operations. Moreover, the BGN scheme restricts the message size to a single bit or a short-length message, rather than allowing messages of the size of the composite-order group. Other cryptosystems, such as Paillier [12], GM [9], [11], Rabin [14], and RSA [15], also operate within composite groups but provide limited homomorphic properties.

The Rabin cryptosystem [14] stands out for its simplicity and efficiency among composite-group schemes. Grounded in the hardness of the factoring problem, it provides a one-way encryption mechanism where the ciphertext is computed as the square of the plaintext modulo a composite integer.

This paper presents a novel homomorphic encryption scheme that builds upon the Rabin cryptosystem to achieve one-way homomorphic properties within composite groups. Its key contributions include a new encryption framework that efficiently encrypts two messages within a single ciphertext, the definition and validation of four homomorphic operations (scalar multiplication, scalar addition, ciphertext multiplication, and ciphertext addition), a security analysis linking the scheme's one-wayness to the factoring problem, and the establishment of a foundational approach that paves the way for constructing more advanced, potentially Indistinguishability under Chosen-Plaintext Attack (IND-CPA) secure cryptosystems.

2 Background

This section introduces the key mathematical tools that facilitate our scheme's design and analysis, setting the stage for understanding its operation within composite groups.

2.1 Quadratic residue

Quadratic residues play a significant role in cryptosystems based on composite groups, such as the Rabin cryptosystem. If p is an odd prime, then half of the residue classes that are relatively prime to p are called *quadratic residues*; these correspond to the integers that are *perfect squares modulo* p. Since $a^2 \equiv (-a)^2 \pmod{p}$ and $a \not\equiv -a \pmod{p}$ for $a \neq 0$, each non-zero quadratic residue has exactly *two square roots modulo* p, differing only in sign.

The structure becomes more intricate when considering a composite modulus $N = pq$, where p and q are two distinct odd primes. Let $k \in \mathbb{Z}_N$ be an integer such that:

$$k_p \equiv k \pmod{p}, \quad k_q \equiv k \pmod{q}$$

By the *Chinese Remainder Theorem (CRT)*, k can be reconstructed using the formula:

$$k = \texttt{CRT}(k_p, k_q) = k_p \cdot q \cdot q' + k_q \cdot p \cdot p' \pmod{N}$$

where $q' \equiv q^{-1} \pmod{p}$ and $p' \equiv p^{-1} \pmod{q}$ are modular inverses.
Furthermore, defining:

$$k_1 = \texttt{CRT}(k_p, k_q), \quad k_2 = \texttt{CRT}(k_p, -k_q), \quad k_3 = \texttt{CRT}(-k_p, k_q), \quad k_4 = \texttt{CRT}(-k_p, -k_q),$$

it follows that:

$$k_1^2 \equiv k_2^2 \equiv k_3^2 \equiv k_4^2 \pmod{N}.$$

Thus, every *quadratic residue modulo* N, when relatively prime to N, has exactly *four square roots modulo* $N = pq$.

2.2 Rabin Cryptosystem

Rabin's scheme requires selecting prime numbers p and q such that $p \equiv q \equiv 3\ (mod\ 4)$, enabling efficient computation of square roots in $\mathbb{Z}_N$, where $N = p \cdot q$. In the Rabin cryptosystem, encryption is performed by squaring the plaintext message m, producing the ciphertext:

$$c = m^2\ (mod\ N)$$

Decryption involves computing the square root of c modulo N using the following equations:

$$m_p = \pm\sqrt{c} \equiv \pm c^{\frac{p+1}{4}}\ (\text{mod}\ p) \tag{1}$$

$$m_q = \pm\sqrt{c} \equiv \pm c^{\frac{q+1}{4}}\ (\text{mod}\ q) \tag{2}$$

The plaintext message m is then recovered using CRT as $m = \texttt{CRT}(m_p, m_q)$.

Since the decryption process results in four possible values, only one of which corresponds to the original plaintext, the probability of successful decryption is $\frac{1}{4}$ in the Rabin cryptosystem. Additionally, Rabin encryption satisfies one-way security but is not IND-CPA secure, making it vulnerable to chosen-plaintext attacks.

3 Proposed Scheme

In this section, we introduce a homomorphic encryption scheme adapted from the Rabin cryptosystem [14]. The scheme consists of four core algorithms: `KeyGen`, `Enc`, `Dec`, and `Eval`, corresponding to *key generation*, *encryption*, *decryption*, and *evaluation*, respectively. The `Eval` algorithm will be detailed in the next section, while the `KeyGen`, `Enc`, and `Dec` algorithms are defined as follows:

- `KeyGen`: Generate two large, distinct prime numbers p and q such that $p \equiv q \equiv 3\ (\text{mod}\ \ 4)$, and compute the public key $pk = N = p \cdot q$. The private key is $sk = (p, q)$, enabling factorization of N for decryption.

- `Enc`: For a (nonzero) message $m \in \mathbb{Z}_N$, select a random integer $r \in \mathbb{Z}_N$ and generate the ciphertext as

$$ctx = \{S,\ P\} \equiv \{m + r,\ mr\}\ (\text{mod}\ N).$$

- `Dec`: Given a fresh ciphertext $ctx = \{S,\ P\}$, decrypt by solving the quadratic equation:

$$x^2 - Sx + P = 0\ (\text{mod}\ \ N).$$

To solve this equation, first compute:

$$\delta = \sqrt{\Delta} = \sqrt{S^2 - 4P}\ (\text{mod}\ \ N)$$

which yields four possible roots $\delta_i; i \in [1,4]$. Then compute:

$$x_i = \frac{S + \delta_i}{2}; i \in [1,4]$$

Among these roots, x_1 corresponds to the original message m, x_2 corresponds to the random integer r, and x_3 and x_4 are additional roots. As a result, the probability of successful decryption is $\frac{1}{4}$.

Remark 1. Since the random integer r can be recovered in the same way as the message m, it can be treated as a second message. As a result, the ciphertext effectively encrypts two messages simultaneously, forming what is known as a Double Message Ciphertext.

3.1 Evaluation

This section introduces four primary evaluation operations: *scalar multiplication*, *scalar addition*, *ciphertext multiplication*, and *ciphertext addition*. While scalar operations do not affect the ciphertext size or the decryption probability, ciphertext operations lead to an increase in ciphertext size and a decrease in decryption probability. The following formally define these operations and establish their homomorphic properties.

Scalar Operations

1. Scalar Multiplication For a ciphertext $ctx = \{S, P\} = \{m + r, mr\}$ and a constant $c \in \mathbb{Z}_N$, the scalar multiplication `SMult` is computed as:

$$ctx_{SM} = \texttt{SMult}(c, ctx) = \{cS,\ c^2P\} = \{cm + cr,\ (cm)(cr)\} \tag{3}$$

Since $cr \in \mathbb{Z}_N$ is a random integer, the result is equivalent to $Enc(cm)$.

2. Scalar Addition : For the same ctx and constant $c \in \mathbb{Z}_N$, compute the scalar addition `SAdd` as:

$$\begin{aligned} ctx_{SA} = \texttt{SAdd}(c, ctx) &= \{S + 2c,\ P + cS + c^2\} \\ &= \{m + r + 2c,\ mr + cm + cr + c^2\} \\ &= \{(m + c) + (r + c),\ (m + c)(r + c)\} \end{aligned} \tag{4}$$

Since $r + c \in \mathbb{Z}_N$ is a random integer, the result is equivalent to $Enc(m + c)$.

Ciphertext Operations

1. Ciphertext Multiplication For two ciphertexts $ctx_1 = \{m_1 + r_1,\ m_1 r_1\}$ and $ctx_2 = \{m_2 + r_2,\ m_2 r_2\}$, multiplication, which is shown by `Mult`, is done as follows:

$$ctx_{CM} = \texttt{CMult}(ctx_1, ctx_2) = \{\ S_1 S_2\ ,\ (S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1\ ,\ P_1 P_2\ \}$$

The result is a *nested ciphertext* with three components:

$$ctx_{CM} = Enc_{\times}^2(m_1.m_2) = \{S_{\times} + r, S_{\times} \cdot r, P_{\times}\},$$

Where

$$S_{\times} = m_1 m_2 + r_1 r_2, \quad r = m_1 r_2 + m_2 r_1, \quad P_{\times} = m_1 m_2 r_1 r_2.$$

Following, the correctness of the first component of ctx_{CM} is demonstrated.

$$\begin{aligned} S_1 S_2 &= (m_1 + r_1)(m_2 + r_2) = m_1 m_2 + m_1 r_2 + r_1 m_2 + r_1 r_2 \\ &= (m_1 m_2 + r_1 r_2) + (m_1 r_2 + m_2 r_1) = S_{\times} + r \end{aligned}$$

The correction of the second part is also as follows:

$$\begin{aligned} &(S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1 \\ &\qquad = ((m_1 + r_1)^2 - 2m_1 r_1)m_2 r_2 + ((m_2 + r_2)^2 - 2m_2 r_2)m_1 r_1 \\ &\qquad = (m_1^2 + r_1^2)m_2 r_2 + (m_2^2 + r_2^2)m_1 r_1 \\ &\qquad = (m_1 m_2 + r_1 r_2)(m_1 r_2 + m_2 r_1) = S_{\times} \cdot r \end{aligned} \tag{5}$$

The third component of ctx_{CM} is as follows:

$$P_1 P_2 = (m_1 r_1)(m_2 r_2) = (m_1 m_2)(r_1 r_2) = P_{\times} \tag{6}$$

To decrypt a nested ciphertext $Enc_{\times}^2(m_1, m_2)$, begin by parsing the first two components as a fresh ciphertext $Enc(S_{\times}) = \{S_{\times} + r,\ S_{\times} \cdot r\}$, where $r = m_1 r_2 + m_2 r_1$ is a random integer. Run the decryption algorithm on $Enc(S_{\times})$ to obtain $S_{\times}$. Then construct the fresh ciphertext $\{S_{\times}, P_{\times}\} = \{m_1 m_2 + r_1 r_2, m_1 m_2 r_1 r_2\}$, which is equivalent to $Enc(m_1 m_2)$ because $r_1 r_2$ is also a random integer. Finally, run the decryption algorithm on $Enc_{\times}(m_1, m_2)$ to obtain the plaintext $m_1 m_2 = m_{\times}$.

Note that the size of the ciphertext grows with each evaluation operation, and the probability of successful decryption for the nested ciphertext decreases to $(\frac{1}{4})^2 = \frac{1}{16}$.

2. Ciphertext Addition For two ciphertexts $ctx_1 = \{m_1 + r_1,\ m_1 r_1\}$ and $ctx_2 = \{m_2 + r_2,\ m_2 r_2\}$, the addition operation, denoted by `CAdd`, is computed as:

$$\begin{aligned} ctx_{CA} &= \texttt{CAdd}(ctx_1, ctx_2) \\ &= \big\{S_1 + S_2, \\ &\quad S_1 \cdot S_2 + 2(P_1 + P_2), \\ &\quad (P_1 + P_2)(S_1 S_2) + (P_1 + P_2)^2 \quad + (S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1\big\} \end{aligned} \tag{7}$$

Like in the ciphertext multiplication evaluation, the result is a *nested ciphertext* with three components:

$$ctx_{CA} = Enc_+^2(m_1 + m_2) = \{S_+, P_+ + r, P_+ \cdot r\}$$

where:

$$S_+ = (m_1+m_2)+(r_1+r_2), \quad P_+ = (m_1+m_2)(r_1+r_2), \quad r = (m_1+r_2)(m_2+r_1).$$

In the following, the investigation and the correctness of each of these components are demonstrated.

$$S_1 + S_2 = (m_1 + r_1) + (m_2 + r_2) = (m_1 + m_2) + (r_1 + r_2)$$

The correction of the second part also as follows:

$$\begin{aligned} 2(P_2 + P_1) + S_1.S_2 &= 2(m_1 r_1 + m_2 r_2) + (m_1 + r_1)(m_2 + r_2) \\ &= 2m_1 r_1 + 2m_2 r_2 + m_1 m_2 + m_1 r_2 + r_1 m_2 + r_1 r_2 \\ &= (m_1 r_1 + m_1 m_2 + m_2 r_2 + m_2 r_1) + (m_1 r_1 + m_2 r_2 + m_1 m_2 + r_1 r_2) \\ &= (m_1 + m_2)(r_1 + r_2) + (m_1 + r_2)(m_2 + r_1) = P_+ + r \end{aligned}$$

Third component of ctx_{CA} is as follows:

$$\begin{aligned} P_+ \cdot r &= (P_1 + P_2)(S_1 S_2) + (P_1 + P_2)^2 + (S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1 \\ &= (m_1 r_1 + m_2 r_2)(m_1 + r_1)(m_2 + r_2) + (m_1 r_1 + m_2 r_2)^2 \\ &\quad + (m_1^2 + r_1^2) m_2 r_2 + (m_2^2 + r_2^2) m_1 r_1 \\ &= \cdots = (m_1 + m_2)(r_1 + r_2) \cdot (m_1 + r_2)(m_2 + r_1) = P_+ \cdot r \end{aligned}$$

To decrypt a nested ciphertext $Enc_+^2(m_1 + m_2)$, begin by parsing the last two components as a fresh ciphertext $Enc(P_+) = \{P_+ + r,\ P_+ \cdot r\}$, where $r = (m_1 + r_2)(m_2 + r_1)$ is a random integer. Next, run the decryption algorithm to obtain P_+. Then construct the fresh ciphertext $\{S_+, P_+\} = \{(m_1 + m_2) + (r_1 + r_2),\ (m_1+m_2)(r_1+r_2)\}$, which is equivalent to $Enc(m_1+m_2)$ because r_1+r_2 is also a random integer. Finally, run the decryption algorithm on $Enc(m_1 + m_2)$ to obtain the plaintext $m_1 + m_2 = m_+$.

4 Improvements

This section presents several enhancements to the homomorphic encryption scheme, focusing on improving the decryption success probability and expanding evaluation capabilities. These improvements address the limitations of the basic scheme, which supports only one addition or one multiplication per fresh ciphertext and has a decryption success probability of $\frac{1}{4}$.

4.1 Probability Enhancement

Efforts to improve decryption success probabilities in the Rabin cryptosystem, such as those in [10,17], could potentially be adapted to our scheme. However, these techniques typically apply only to fresh ciphertexts, losing effectiveness after evaluation operations. To address this, we introduce modifications designed to elevate the probability of successful decryption across both fresh and nested ciphertexts.

Fresh ciphertext Let a fresh ciphertext be represented as $ctx = \{S, P\} = \{m + r, mr\}$, where $m \in \mathbb{Z}_N$ is the message and $r \in \mathbb{Z}_N$ is a random integer. To enhance decryption reliability, we select r such that $m - r$ is a perfect square. Specifically, given m, we choose a random integer $\sigma \in \mathbb{Z}_N$, such that $\sigma^2 < N$, so $\sigma^2 = \sigma \cdot \sigma \mod N$ is a perfect square, and set $r = m - \sigma^2 \mod N$. This preserves the randomness of r due to the random selection of σ. The discriminant δ is then computed as:

$$\delta = \sqrt{\Delta} = \sqrt{S^2 - 4P} = \sqrt{(m+r)^2 - 4mr} = \sqrt{(m-r)^2} \tag{8}$$

Since $m - r = \sigma^2$, this implies $\delta = \sqrt{(\sigma^2)^2} = |\sigma^2| \mod N$, and thus two of the four possible roots, δ_1 and δ_2, are $\pm\sigma^2$. The other two roots, δ_3 and δ_4, depend on the modulo N structure and are typically not perfect squares.

The decryption algorithm examines all four potential roots, $\delta_1, \delta_2, \delta_3$, and δ_4, and identifies which are perfect squares modulo N. Since σ^2 is a perfect square by construction, $\delta_1 = \sigma^2$ satisfies this condition. With high probability, the other roots (including $\delta_2 = -\sigma^2$) are not perfect squares, as -1 is not a quadratic residue modulo N when N's factors p and q satisfy $p, q \equiv 3 \mod 4$ (per Section 2.2) [3]. The message can then be recovered as follows:

$$m = \frac{S + \sigma^2}{2} = \frac{(m+r) + \sigma^2}{2} = \frac{m + (m - \sigma^2) + \sigma^2}{2} \tag{9}$$

This modification raises the decryption success probability to approximately 1. The impact on nested ciphertexts is discussed in the next subsection.

[3] Note that in our setting, $-\sigma^2$ is not a perfect square. One of the two other roots is a quadratic residue in $\mathbb{Z}_N$, but this does not necessarily mean that it is a perfect square integer.

Multiplied Nested Ciphertext Consider two ciphertexts:

$$ctx_1 = \{S_1, P_1\} = \{m_1 + r_1, m_1 r_1\}, \quad r_1 = m_1 - \sigma_1^2$$

$$ctx_2 = \{S_2, P_2\} = \{m_2 + r_2, m_2 r_2\}, \quad r_2 = m_2 - \sigma_2^2$$

For the encryption of their product, $E_\times(m_1 m_2)$, the following holds:

$$\begin{aligned}\{S_\times + r,\ S_\times \cdot r,\ P_\times\} &= \{S_1 S_2,\ (S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1,\ P_1 P_2\} \\ &= \{(m_1 m_2 + r_1 r_2) + (m_1 r_2 + m_2 r_1), \\ &\qquad (m_1 m_2 + r_1 r_2)(m_1 r_2 + m_2 r_1),\ (m_1 m_2)(r_1 r_2)\}\end{aligned}$$

To decrypt this nested ciphertext, we first treat $\{S_\times + r, S_\times \cdot r\}$ as a fresh ciphertext and proceed with decryption. The decryption algorithm computes the following:

$$\begin{aligned}&\sqrt{(S_\times + r)^2 - 4(S_\times \cdot r)} = \sqrt{(S_\times - r)^2} \mod N \\ &= \sqrt{((m_1 m_2 + r_1 r_2) - (m_1 r_2 + m_2 r_1))^2} \\ &= \sqrt{(m_1 m_2 + (m_1 - \sigma_1^2)(m_2 - \sigma_2^2) - m_1(m_2 - \sigma_2^2) - m_2(m_1 - \sigma_1^2))^2} \\ &= \sqrt{(m_1 m_2 + m_1 m_2 - m_1\sigma_1^2 - m_2\sigma_1^2 + \sigma_1^2\sigma_2^2 - m_1(m_2 - \sigma_2^2) - m_2(m_1 - \sigma_1^2))^2} \\ &= \sqrt{(2m_1 m_2 - m_1\sigma_1^2 - m_2\sigma_1^2 + \sigma_1^2\sigma_2^2 - m_1 m_2 + m_1\sigma_2^2 - m_2 m_1 + m_2\sigma_1^2))^2} \\ &= \sqrt{(\sigma_1^2\sigma^2)^2} \mod N\end{aligned}$$

Thus, one of the solutions to the above equation is $\sigma_1^2\sigma_2^2 \mod N$. However, it is important to note that $\sigma_1^2\sigma_2^2 \mod N$ is not necessarily a perfect square integer. This is because $\sigma_1^2\sigma_2^2$ may exceed N, leading to an overflow. Nevertheless, it is always a quadratic residue modulo N.

Since the result of $\sqrt{(\sigma_1^2\sigma_2^2)^2} \mod N$ has two quadratic residue roots, the probability of correctly determining $\sigma_1^2\sigma_2^2$ is $\frac{1}{2}$.

Added Nested Ciphertext Consider two ciphertexts:

$$ctx_1 = \{S_1, P_1\} = \{m_1 + r_1, m_1 r_1\}, \quad r_1 = m_1 - \sigma_1^2$$

$$ctx_2 = \{S_2, P_2\} = \{m_2 + r_2, m_2 r_2\}, \quad r_2 = m_2 - \sigma_2^2$$

For the encryption of their summation, $E_+(m_1 + m_2)$, the following holds:

$$\begin{aligned}\{S_+,\ P_+ + r,\ P_\times \cdot r\} &= \{S_1 + S_2,\ S_1S_2 + 2(P_2 + P_1),\\ &\quad (P_1 + P_2)(S_1S_2) + (P_1 + P_2)^2 + (S_1^2 - 2P_1)P_2 + (S_2^2 - 2P_2)P_1\}\\ &= \{(m_1 + m_2) + (r_1 + r_2),\\ &\quad (m_1 + m_2)(r_1 + r_2) + (m_1 + r_2)(m_2 + r_1),\\ &\quad (m_1 + m_2)(r_1 + r_2) \cdot (m_1 + r_2)(m_2 + r_1)\ \ \}\end{aligned}$$

To decrypt this nested ciphertext, we first treat $\{P_+ + r,\ P_\times \cdot r\}$ as a fresh ciphertext and proceed with decryption. The decryption algorithm computes the following:

$$\begin{aligned}&\sqrt{(P_+ + r)^2 - 4(P_+ \cdot r)} = \sqrt{(P_+ - r)^2} \mod N\\ &= \sqrt{((m_1 + m_2)(r_1 + r_2) - (m_1 + r_2)(m_2 + r_1))^2}\\ &= \sqrt{(m_1r_1 + m_1r_2 + m_2r_1 + m_2r_2 - (m_1m_2 + m_1r_1 + m_2r_2 + r_1r_2))^2}\\ &= \sqrt{(m_1r_2 + m_2r_1 - m_1m_2 - r_1r_2)^2} = \sqrt{((m_1 - r_1)(r_2 - m_2))^2}\\ &= \sqrt{(m_1 - r_1)^2(r_2 - m_2)^2} = \sqrt{(\sigma_1^2\sigma^2)^2} \mod N\end{aligned}$$

Thus, one of the solutions to the above equation is $\sigma_1^2\sigma_2^2 \mod N$. However, it is important to note that $\sigma_1^2\sigma_2^2 \mod N$ is not necessarily a perfect square integer. This is because $\sigma_1^2\sigma_2^2$ may exceed N, leading to an overflow. Nevertheless, it is always a quadratic residue modulo N.

Since the result of $\sqrt{(\sigma_1^2\sigma_2^2)^2} \mod N$ has two quadratic residue roots, the probability of correctly determining $\sigma_1^2\sigma_2^2$ is $\frac{1}{2}$.

Scalar operations The proposed technique is not effective for scalar operations, i.e., scalar multiplication and scalar addition. However, if the constant c in scalar multiplication is a quadratic residue, the decryption probability grows to $\frac{1}{2}$. This is because of the following property for a ciphertext $\{cS,\ c^2P\}$.

$$\delta = \sqrt{\Delta} = \sqrt{c^2S^2 - 4c^2P} = \sqrt{c^2(m + r)^2 - 4c^2mr} = \sqrt{c^2(m - r)^2} = \sqrt{(c\sigma^2)^2}$$

This means that one of the roots is $c\sigma^2$ which is quadratic residue by assumption. Therefore, the probability of detecting this value is $\frac{1}{2}$, as there is another root that is also a quadratic residue.

4.2 Interactive Evaluation

The proposed scheme supports either one multiplication or one addition operation using ciphertext multiplication and ciphertext addition, respectively. However, this is insufficient for evaluating more complex functions. To address this limitation, this section introduces an interactive evaluation technique that enables the evaluation of more complex functions. In this approach, the data user

(i.e., the destination party holding the secret key) must be online and collaborate with the evaluator, while ensuring that intermediate data remains hidden from the data user.

Consider a nested ciphertext of the form:

$$E_{\times}(m_1 m_2) = \{S_{\times} + r,\ S_{\times} \cdot r,\ P_{\times}\}$$

The evaluator treats $\{S_{\times} + r,\ S_{\times} \cdot r\}$ as a fresh ciphertext. To facilitate secure interactive decryption, the evaluator performs the following steps:

1. Randomization: The evaluator selects a random integer $c \in \mathbb{Z}_N$ and performs a scalar multiplication on the ciphertext:

$$\{c(S \times +r),\ c^2 \cdot S_{\times} \cdot r\}$$

   This randomized ciphertext is then sent to the data user.
2. Decryption by the Data User: The data user applies the decryption algorithm to recover $cS_{\times}$ and forwards this value to the evaluator, who can then compute cr. To preserve security, the data user must not reveal other possible decryption solutions, as this could allow the evaluator to factor N.
3. Recovery of $S_{\times}$: Upon receiving $cS_{\times}$, the evaluator removes the random scalar c and computes $S_{\times}$. The ciphertext is then updated to $S_{\times}, P_{\times}$, which serves as a fresh ciphertext, allowing the evaluation process to continue.

It is important that the data user sends the same value of $S_{\times}$ consistently, which occurs with a probability of $\frac{1}{4}$. This probability can be increased to $\frac{1}{2}$ if the encryption algorithm follows the technique described in Section 4.1 and the evaluator ensures that the chosen constant c is a quadratic residue. To guarantee this, the evaluator selects a random integer $t \in \mathbb{Z}_N$ and computes $c = t^2 \mod N$.

A similar interactive decryption process can be applied to a nested ciphertext of the form:

$$E_{+}(m_1 + m_2) = \{S_{+},\ P_{+} + r,\ P_{\times} \cdot r\}$$

In this case, the evaluator extracts $\{S_{+},\ P_{+} + r\}$ and performs the following steps:

1. Randomization: The evaluator selects a random integer $c \in \mathbb{Z}_N$ and performs a scalar multiplication on the ciphertext:

$$\{c(P + +r), c^2 \cdot P_{+} \cdot r\}$$

   This randomized ciphertext is then sent to the data user.
2. Decryption by the Data User: The data user applies the decryption algorithm to recover cP_{+} and forwards this value to the evaluator.
3. Recovery of P_{+}: Upon receiving cP_{+}, the evaluator removes the random scalar c and computes P_{+}. The ciphertext is then updated to S_{+}, P_{+}, which serves as a fresh ciphertext, allowing the evaluation process to continue.

The probability of returning the correct value cP_{+} is $\frac{1}{4}$, which can be increased to $\frac{1}{2}$ by following the technique described in Section 4.1 and ensuring that c is a quadratic residue, computed as $c = t^2 \mod N$.

5 Analysis

This section provides an analysis of the proposed scheme, focusing on its decryption probability and security properties. Key remarks and a formal theorem establish the scheme's strengths and limitations, linking its security to established cryptographic assumptions.

First, the decryption probability of nested ciphertexts, such as $Enc^2_{\times}(m_1, m_2)$ or $Enc^2_{+}(m_1+m_2)$, is $\frac{1}{16}$. This is because two sequential decryptions are required: one to recover the intermediate value (e.g, $S_{\times}$ or P_{+}) and the other to obtain the final result (e.g., m_1m_2 or m_1+m_2). Since these steps are independent and each one has a success probability of $\frac{1}{4}$, the total probability of successful decryption is $\frac{1}{4} \times \frac{1}{4} = \frac{1}{16}$. Moreover, even an adversary with knowledge of the secret key or the ability to factor N cannot recover the original messages m_1 and m_2 from the nested ciphertext.

Remark 2. For any adversary $\mathcal{A}$ capable of breaking the Rabin cryptosystem [14], there exists an adversary $\mathcal{N}$ who can factor the public key $N = p \cdot q$.

Theorem 1. *For any adversary $\mathcal{A}$ who can break the proposed cryptosystem, there exists an adversary $\mathcal{B}$ who can break the Rabin cryptosystem.*

Proof. Let c be a ciphertext in the Rabin cryptosystem, computed as $c = m^2 \mod N$, with public key N. Adversary $\mathcal{B}$ constructs a ciphertext in the proposed scheme as follows: select a random integer $S \in \mathbb{Z}_N$, compute

$$P = \frac{S^2 - c}{4} \mod N,$$

and form $ctx = \{S, P\}$. This ctx satisfies the structure of the proposed encryption, since $S = x_1 + x_2$ and $P = x_1x_2$ for roots x_1 and x_2 of the equation

$$x^2 - Sx + P = 0 \mod N,$$

where

$$(x_1 - x_2)^2 = S^2 - 4P = c.$$

Adversary $\mathcal{B}$ submits ctx to $\mathcal{A}$, who, if capable of breaking the proposed scheme, returns

$$x_i = \frac{S + \delta_i}{2} \quad \text{for} \quad i \in [1, 4],$$

where

$$\delta_i = \sqrt{S^2 - 4P} = \sqrt{c} \mod N.$$

From this, $\mathcal{B}$ computes

$$\delta_i = 2x_i - S,$$

recovering a square root of c. Since computing square roots modulo N in the Rabin cryptosystem is equivalent to factoring N, $\mathcal{B}$ succeeds whenever $\mathcal{A}$ does, establishing the reduction.

6 Conclusion

This paper introduces a novel cryptosystem that employs innovative techniques to achieve homomorphic properties for four fundamental operations: scalar multiplication, scalar addition, ciphertext multiplication, and ciphertext addition. Building on the Rabin cryptosystem, the proposed scheme supports either a single ciphertext multiplication or a single ciphertext addition, with the ciphertext size increasing by a factor of 1.5 after each such operation. This approach strikes a compelling balance between computational efficiency and functional versatility, offering a new lens on secure computation within composite groups.

While the current design is limited to operations involving two ciphertexts, the proposed techniques have the potential to be extended to support multiple ciphertext multiplications and additions. Additionally, although the current construction only achieves one-way security, we believe it opens avenues for further development toward achieving IND-CPA security through the introduction of randomness and additional structure.

These findings contribute to advancing homomorphic encryption by balancing efficiency, functionality, and security, paving the way for further improvements and practical implementations.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation through the PID2021-125962OB-C31 "SECURING" project. Additional funding was provided by the ARTEMISA International Chair of Cybersecurity (C057/23) and the DANGER Strategic Project of Cybersecurity (C062/23), both funded by the Spanish National Institute of Cybersecurity through the European Union — NextGenerationEU and the Recovery, Transformation, and Resilience Plan.

References

1. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3378, pp. 325–341. Springer (2005). https://doi.org/10.1007/978-3-540-30576-7_18, https://iacr.org/archive/tcc2005/3378_325/3378_325.pdf
2. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: Annual International Cryptology Conference (2012), https://api.semanticscholar.org/CorpusID:9156246
3. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. p. 309–325. ITCS '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2090236.2090262, https://doi.org/10.1145/2090236.2090262

4. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: International Conference on the Theory and Application of Cryptology and Information Security (2017), https://api.semanticscholar.org/CorpusID:3164123
5. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe. In: International Conference on the Theory and Application of Cryptology and Information Security (2017), https://api.semanticscholar.org/CorpusID:40718273
6. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: International Conference on the Theory and Application of Cryptographic Techniques (2010), https://api.semanticscholar.org/CorpusID:5627147
7. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptol. ePrint Arch. **2012**, 144 (2012), https://api.semanticscholar.org/CorpusID:1467571
8. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. p. 169–178. STOC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440, https://doi.org/10.1145/1536414.1536440
9. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing. p. 365–377. STOC '82, Association for Computing Machinery, New York, NY, USA (1982). https://doi.org/10.1145/800070.802212, https://doi.org/10.1145/800070.802212
10. Mahad, Z., Asbullah, M., Ariffin, M.: Efficient methods to overcome rabin cryptosystem decryption failure. Malaysian Journal of Mathematical Sciences **11**, 9–20 (2017)
11. Mahdavi, M., Meamari, E., Beni, E.H., Sheikhi, M.: Leveled homomorphic encryption over composite groups. In: International Conference on Cryptology in Africa. pp. 25–50. Springer (2025), https://doi.org/10.1007/978-3-031-97260-7_2
12. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International conference on the theory and applications of cryptographic techniques. pp. 223–238. Springer (1999)
13. Pereira, H.V.L.: Efficient agcd-based homomorphic encryption for matrix and vector arithmetic. In: International Conference on Applied Cryptography and Network Security. pp. 110–129. Springer (2020)
14. Rabin, M.O.: Digitalized signatures and public-key functions as intractable as factorization (1979)
15. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
16. Stehlé, D., Steinfeld, R.: Faster fully homomorphic encryption. In: Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16. pp. 377–394. Springer (2010)
17. Williams, H.: A modification of the rsa public-key encryption procedure (corresp.). IEEE Transactions on Information Theory **26**(6), 726–729 (1980)