Indian Institute of Technology Goa

BTECH Project Report

Goldwasser Micali Cryptosystem

Authors: Devyani Remulkar,Harshini Maryada

Supervisor: Prof. Arpita Korwar

Department of Computer Science and Engineering

November 24, 2025

Abstract

This project examines the Goldwasser–Micali (GM) cryptosystem, the first probabilistic public-key scheme achieving IND-CPA security using the quadratic residuosity assumption. We outline the limitations of the original GM scheme—particularly its one-bit encryption and large ciphertext expansion—and then study the Generalized GM cryptosystem, which supports multi-bit encryption using a modulus of the form $N = P^a Q^b$. The project describes key generation, encryption, and decryption, and reports experimental observations on efficiency and ciphertext expansion. Overall, the work highlights how the generalized scheme extends GM while maintaining strong security properties.

Contents

1 Introduction

Cryptography is the science of designing techniques that allow two or more parties to communicate securely in the presence of adversaries. Its goal is to protect information from unauthorized access, modification, or misuse.

Originally used for secret military communication, cryptography today powers nearly every aspect of digital security, including online banking, messaging apps, digital signatures, blockchain, privacy-preserving computation, and secure authentication systems.

1.1 Basic Definitions in Cryptography

- **Plaintext**
  The original message or data that needs protection.

- **Ciphertext**
  The encrypted form of the plaintext, produced by an encryption algorithm. It should appear random to an attacker.

- **Key**
  A piece of secret information used by encryption and decryption algorithms. The security of a cryptosystem relies on the secrecy of the key, not the algorithm.

- **Encryption**
  The process of transforming plaintext into ciphertext using a key.

$$\text{Enc}_K(M) = C$$

- **Decryption**
  The process of recovering plaintext from ciphertext using a key.

$$\text{Dec}_K(C) = M$$

1.2 Symmetric-Key (Private-Key) Cryptography

Symmetric-key cryptography uses the same secret key for both encryption and decryption. It offers several advantages:

- It is computationally very fast and efficient.

- It requires minimal computational resources.

However, it has a fundamental drawback: both communicating parties must share the same secret key *securely*. Without a secure channel or prior trust, exchanging this secret key becomes difficult. This limitation motivated the development of public-key cryptography.

1.3 Public Key Cryptography (PKC)

Public Key Cryptography (PKC) is a foundational cryptographic primitive introduced to resolve the key distribution problem of symmetric systems. Early PKC systems, such as RSA, were deterministic. As a consequence, they were vulnerable to chosen-plaintext attacks (CPA), because identical messages always produced identical ciphertexts.

1.4 The Goldwasser–Micali Breakthrough

In 1982, Shafi Goldwasser and Silvio Micali introduced the first *probabilistic* public-key cryptosystem, now known as the Goldwasser–Micali (GM) cryptosystem. Their work established a new standard in cryptography by showing that:

- Probabilistic encryption is essential for achieving **semantic security**.
- Semantic security is equivalent to the modern notion of **IND-CPA** (Indistinguishability under Chosen-Plaintext Attack).

1.5 Why Probabilistic Encryption?

For a cryptosystem to achieve IND-CPA security, ciphertexts corresponding to different messages must be indistinguishable. This property cannot be achieved by deterministic encryption, since the same plaintext always yields the same ciphertext.

The GM cryptosystem was the first to implement randomized encryption based on the **quadratic residuosity assumption**, thereby achieving IND-CPA security and laying the foundation for modern secure public-key systems.

2 Preliminaries

2.1 Quadratic Residue

Let N be an integer, typically an odd composite modulus in cryptographic settings. An integer a is called a *quadratic residue modulo* N if there exists some x such that

$$x^2 \equiv a \pmod{N}.$$

If no such x exists, a is called a *quadratic non-residue.*

2.2 Jacobi Symbol and Its Properties

For an odd prime number $p > 2$, the Jacobi symbol reduces to a simple quadratic test. It is defined as

$$J_p(a) = \left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p \mid a, \\ 1, & \text{if } a \text{ is a quadratic residue modulo } p, \\ -1, & \text{if } a \text{ is a quadratic non-residue modulo } p. \end{cases}$$

Moreover, for primes, the Jacobi symbol can be computed using the formula

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p},$$

which follows from Euler's criterion. This identity provides an efficient way to check whether a behaves like a residue modulo a prime.

For an odd composite modulus of the form

$$N = p_1 p_2 \cdots p_k,$$

the Jacobi symbol is extended multiplicatively:

$$J_N(a) = \left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_k}\right).$$

An important fact is that an integer a is a quadratic residue modulo N if and only if it is a quadratic residue modulo each of the prime factors of N.

Even though its definition involves the prime factors of N, the Jacobi symbol can be computed efficiently without knowing the factorization, using recursive modular reduction rules.

Jacobi Symbol Ambiguity in Composite Moduli

When the modulus is prime, the Legendre symbol provides a perfect indicator of quadratic residuosity:

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

However, for a composite modulus $N = pq$, the Jacobi symbol is defined as:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right).$$

This value does **not provide a definitive guarantee** about whether a is a quadratic residue modulo N.

Case 1: $\left(\frac{a}{N}\right) = -1$

If the Jacobi value is -1, then at least one of the values $\left(\frac{a}{p}\right)$ or $\left(\frac{a}{q}\right)$ is -1. Thus:

$$\left(\frac{a}{N}\right) = -1 \quad \Rightarrow \quad a \text{ is a quadratic non-residue modulo } N.$$

So, this case behaves similarly to the prime modulus case.

Case 2: $\left(\frac{a}{N}\right) = 1$

This is the **ambiguous** case. It can occur in two situations:

$$\left(\frac{a}{p}\right) = 1, \left(\frac{a}{q}\right) = 1 \quad \Rightarrow \quad a \text{ is a true quadratic residue modulo } N,$$

$$\left(\frac{a}{p}\right) = -1, \left(\frac{a}{q}\right) = -1 \quad \Rightarrow \quad a \text{ is a quadratic non-residue modulo } N.$$

Therefore:

$$\left(\frac{a}{N}\right) = 1 \not\Rightarrow a \text{ is a quadratic residue modulo } N.$$

In fact, among all values a such that $\left(\frac{a}{N}\right) = 1$, approximately:

$$50\% \text{ are quadratic residues, and } 50\% \text{ are non-residues.}$$

This property is exploited in the Goldwasser–Micali cryptosystem, where all ciphertexts are constructed to have Jacobi symbol 1. Without the prime factors of N, an adversary cannot distinguish between residues and non-residues, providing the indistinguishability required for semantic security.

2.3 Quadratic Residuosity Assumption

The indistinguishability between quadratic residues (QR) and quadratic non-residues (QNR) that have Jacobi symbol 1 forms the computational basis of the Goldwasser–Micali cryptosystem. This hardness is formalized by the **Quadratic Residuosity Assumption (QRA)**. The assumption states that if $N = pq$ is a large composite modulus, it is computationally infeasible for any polynomial-time adversary to determine whether a given element x with $J_N(x) = 1$ is a true quadratic residue modulo N or a non-residue. In other words, without the trapdoor information—the factorization (p, q)—non-residues that appear indistinguishable from residues cannot be efficiently separated from actual residues. The security of the Goldwasser–Micali cryptosystem relies entirely on this presumed hardness.

3 Goldwasser–Micali (GM) Cryptosystem

3.1 Introduction

The Goldwasser–Micali cryptosystem is the first public-key encryption scheme proven to achieve semantic security under the Quadratic Residuosity Assumption. The scheme encrypts each message bit individually and relies on the hardness of distinguishing quadratic residues from non-residues modulo a composite number.

3.2 Key Generation

To generate keys, choose two large distinct odd primes p and q and compute $N = pq$. Select a non-residue $y \in \mathbb{Z}_N^*$ such that

$$\left(\frac{y}{N}\right) = 1.$$

This y is called a *quadratic non-residue with Jacobi symbol* 1. The public key is (N, y) and the secret key is the factorization (p, q).

3.3 Encryption

To encrypt a single bit $m \in \{0, 1\}$, choose a random $r \in \mathbb{Z}_N^*$ and compute the ciphertext

$$c = \begin{cases} r^2 \bmod N, & \text{if } m = 0, \\ yr^2 \bmod N, & \text{if } m = 1. \end{cases}$$

Both ciphertext forms have Jacobi symbol 1, which ensures that the adversary cannot distinguish between them without knowing the factorization of N.

3.4 Decryption

Using the secret key (p, q), the receiver determines whether c is a quadratic residue modulo N. If c is a residue, the decrypted bit is $m = 0$; otherwise, $m = 1$.

If $m = 0$, then $c = r^2$ is always a quadratic residue. If $m = 1$, then $c = yr^2$ is a product of a non-residue (with Jacobi symbol 1) and a square, thus a non-residue. Hence decryption correctly recovers the plaintext bit.

3.5 Security Intuition

The GM scheme is probabilistic: each bit is encrypted using fresh randomness, so encrypting the same bit multiple times yields different ciphertexts. This randomness, combined with the fact that both ciphertext forms have Jacobi symbol 1, ensures that an adversary cannot distinguish encryptions of 0 and 1. The semantic security of the scheme directly relies on the Quadratic Residuosity Assumption.

4 Problems with the GM Cryptosystem

While the Goldwasser–Micali cryptosystem is an important theoretical construction, it is not practical for real-world deployment. Its major issue is the very large ciphertext expansion. Since GM encrypts only a single bit at a time, each ciphertext becomes a full element of $\mathbb{Z}_N^*$, which is roughly $|N|$ bits long. Thus, the ciphertext expansion ratio is approximately $|N| : 1$, meaning that encrypting one plaintext bit produces a ciphertext of size $|N|$ bits. Moreover, each bit requires generating a fresh random square or a random non-square with Jacobi symbol 1, further increasing cost.

This motivates the development of more optimized GM-type schemes with better efficiency.

5 Generalized GM Cryptosystem

5.1 Introduction

An extended version of the classic Goldwasser–Micali (GM) encryption scheme was proposed by Ying Guo, Xiao-Lei Dong, and Zhen-Fu Cao in their 2022 paper titled *"Generalized Goldwasser and Micali's Type Cryptosystem"*, published in the *Journal of Computer Science and Technology.*

The generalized scheme supports **multi-bit encryption**, in contrast to the original GM scheme which encrypts only a single bit per ciphertext.

The construction uses a modulus of the form

$$N = P^a Q^b,$$

where P and Q are distinct primes and $a, b \in \mathbb{N}$, allowing the encoding of larger message blocks within a single ciphertext.

The scheme consists of the following three components:

- **Key Generation**
- **Encryption**
- **Decryption**

5.2 Key Generation

Here we provide some definitions followed by the algorithm.

Definition 1 (κ-Acceptable N). Supposing P, Q are large primes and both $P - 1$ and $Q - 1$ contain at least one large prime factor whose length is at least κ, if

$$N = P^a Q^b$$

and a, b are natural numbers, then we say N is κ-acceptable.

Definition 2 (κ-Acceptable Tuple (N, k)). Supposing N is κ-acceptable, i.e.,

$$N = P^a Q^b,$$

where

$$P-1=\prod_{i=0}^{t} p_i^{u_i} P', \qquad Q-1=\prod_{i=0}^{t} p_i^{v_i} Q',$$

and

$$k_1=\prod_{i=0}^{t} p_i^{s_i} \quad \text{or} \quad k_1=1,$$

where

$$s_i=\begin{cases} \max(u_i,v_i), & \text{if } 1\le i\le t,\\ \max(u_i,v_i), & \text{if } i=0,\ 2\mid(a+b),\\ u_i, & \text{if } i=0,\ 2\mid a,\ 2\nmid b,\\ v_i, & \text{if } i=0,\ 2\nmid a,\ 2\mid b. \end{cases}$$

We say that the tuple (N,k) is κ-acceptable if

$$k=k_1P^{a-1}Q^{b-1}.$$

Definition 3 (Definition of Set $Y(N,k)$). Suppose the tuple (N,k) is κ-acceptable, i.e.,

$$N=P^aQ^b \qquad \text{and} \qquad k=k_1P^{a-1}Q^{b-1},$$

where

$$k_1=\prod_{i=0}^{t} p_i^{s_i}.$$

We define the sets Y_P, Y_Q, and Y_i $(0\le i\le t)$ as follows:

$$Y_P=\begin{cases} \{\, y \mid y^{(P-1)P^{a-2}}\not\equiv 1 \pmod{P^a}\,\}, & \text{if } a>1,\\ \{1\}, & \text{if } a=1. \end{cases}$$

$$Y_Q=\begin{cases} \{\, y \mid y^{(Q-1)Q^{b-2}}\not\equiv 1 \pmod{Q^b}\,\}, & \text{if } b>1,\\ \{1\}, & \text{if } b=1. \end{cases}$$

If $k_1=\prod_{i=0}^{t} p_i^{s_i}$, then for $i=0$, we set

$$Y_0=\begin{cases} \left\{\, y \mid \left(\frac{y}{N}\right)=1,\ \left(\frac{y}{P}\right)=-1\right\}, & \text{if } s_0=u_0,\\ \left\{\, y \,\middle|\, \left(\frac{y}{N}\right)=1,\ \left(\frac{y}{Q}\right)=-1\right\}, & \text{if } s_0=v_0. \end{cases}$$

For $1\le i\le t$, we set

$$Y_i=\begin{cases} \{\, y \mid y^{(P-1)/p_i}\not\equiv 1 \pmod{P}\,\}, & \text{if } s_i=u_i,\\ \{\, y \mid y^{(Q-1)/p_i}\not\equiv 1 \pmod{Q}\,\}, & \text{if } s_i=v_i. \end{cases}$$

We define the set $Y(N,k)$ as

$$Y(N,k)=Y_P\cap Y_Q\cap\bigcap_{i=0}^{t} Y_i.$$

KeyGen Algorithm

KeyGen(1^κ). The KeyGen algorithm takes the security parameter 1^κ as input and outputs a κ-acceptable tuple (N, k), where

$$N = P^a Q^b \qquad \text{and} \qquad k = k_1 P^{a-1} Q^{b-1}.$$

Let l denote the bit-length of k, and define

$$K = k_1 N.$$

We randomly choose

$$y \in Y(N, k),$$

where $Y(N, k)$ is the set defined in Definition 3.

The public key and secret key are then given by

$$\mathrm{pk} = (N, y, l, K), \qquad \mathrm{sk} = (P, Q, k).$$

5.3 Encryption

Enc$_{pk}(m)$**.** For a message $m \in \{0,1\}^{l-1}$, we randomly choose

$$x \in \mathbb{Z}_N^*,$$

and compute the ciphertext

$$c = y^m x^K \bmod N.$$

5.4 Decryption

Dec(c). The decryption algorithm $\mathrm{Dec}(c, y, k, N)$ is given in Algorithm 1. We give a detailed description here.

Supposing $k_1 = \prod_{i=0}^{t} p_i^{s_i}$, for a specific i and assuming $p_i^{s_i} \mid (P-1)$ without loss of generality, we compute:

$$c_i = c^{\frac{P-1}{p_i^{s_i}}} \bmod P, \qquad y_i = y^{\frac{P-1}{p_i^{s_i}}} \bmod P.$$

This removes the random number x from (1) and leads to the equation:

$$c_i \equiv y_i^{m_i} \pmod{P}. \tag{2}$$

By assumption of our scheme, we have $\mathrm{ord}_P(y_i) = p_i^{s_i}$. Thus (2) is a $(c_i, y_i, p_i^{s_i}, P)$ discrete logarithm problem and we solve it using SDLP1. Therefore:

$$m_i = m \bmod p_i^{s_i} = \mathrm{SDLP1}(c_i, y_i, p_i^{s_i}, P).$$

Similarly, suppose $a > 1$, we compute:

$$c_P = c^{P-1} \bmod P^a, \qquad y_P = y^{P-1} \bmod P^a,$$

which removes the randomness and gives:

$$c_P \equiv y_P^{m_P} \pmod{P^a}. \tag{3}$$

It is easy to see $\mathrm{ord}_{P^a}(y_P) = P^{a-1}$. Thus (3) is a (c_P, y_P, P^{a-1}, P^a) discrete logarithm problem and we solve it using SDLP2:

$$m_P = m \bmod P^{a-1} = \mathrm{SDLP2}(c_P, y_P, P^{a-1}, P^a).$$

If $b > 1$, we compute $m_Q = m \bmod Q^{b-1}$ in the same manner. Since we obtain $m \bmod p_i^{s_i}$ for all $0 \leq i \leq t$, and $m \bmod P^{a-1}$ as well as $m \bmod Q^{b-1}$, we finally compute $m \bmod k$ using the Chinese Remainder Theorem (CRT).

Remark: In our report, we treat SDLP1 and SDLP2 as black-box routines. Their high-level descriptions are as follows:

- SDLP1(c, y, k, P): **Prime-power DLP solver (mod P)**
  - **Input:** $(c, y, k = p^s, P)$ with $\mathrm{ord}_P(y) = k$.
  - **Output:** m such that $c \equiv y^m \pmod{P}$.
- SDLP2(c, y, k, N): **Prime-power DLP solver (mod P^a)**
  - **Input:** $(c, y, k = P^{a-1}, N = P^a)$.
  - **Output:** m such that $c \equiv y^m \pmod{N}$.

Algorithm 1. $\textsc{Dec}(c, y, k, N)$

Input: integer $c \in \mathbb{Z}_N^*$, base y, integer k and modulus N, where the tuple (N, k) is λ-acceptable; let $N = P^a Q^b$, where

$$P - 1 = \prod_{i=0}^{t} p_i^{u_i} P' \quad \text{and} \quad Q - 1 = \prod_{i=0}^{t} p_i^{v_i} Q';$$

let $k = k_1 P^{a-1} Q^{b-1}$, where

$$k_1 = \prod_{i=0}^{t} p_i^{s_i} \quad \text{or} \quad k_1 = 1.$$

Output: $m \in \mathbb{Z}_k$ such that $c \equiv y^m x^K \bmod N$ where $K = k$ or $K = k_1 N$

```
$T \leftarrow \emptyset$;
if $k_1 > 1$ then
  for $i$ from $0$ to $t$ do
      if $s_i = u_i$ then
          $n \leftarrow P$;
      else
          $n \leftarrow Q$;
      end if
      $c_i \leftarrow c^{(n-1)/p_i^{s_i}} \bmod n$, $y_i \leftarrow y^{(n-1)/p_i^{s_i}} \bmod n$;
      $m_i \leftarrow$ SDLP1$(c_i, y_i, p_i^{s_i}, n)$;
      $T \leftarrow T \cup \{(m_i, p_i^{s_i})\}$;
  end for
end if
if $a > 1$ then
  $c_P \leftarrow c^{P-1} \bmod P^a$, $y_P \leftarrow y^{P-1} \bmod P^a$;
  $m_P \leftarrow$ SDLP2$(c_P, y_P, P^{a-1}, P^a)$;
  $T \leftarrow T \cup \{(m_P, P^{a-1})\}$;
end if
if $b > 1$ then
  $c_Q \leftarrow c^{Q-1} \bmod Q^b$, $y_Q \leftarrow y^{Q-1} \bmod Q^b$;
  $m_Q \leftarrow$ SDLP2$(c_Q, y_Q, Q^{b-1}, Q^b)$;
  $T \leftarrow T \cup \{(m_Q, Q^{b-1})\}$;
end if
$m \leftarrow$ CRT$(T)$;
```

5.5 Ciphertext Expansion Analysis

In this subsection, we analyze the ciphertext expansion of the generalized GM-type scheme.

First, we recall the parameters of the scheme. The modulus N of our scheme is κ-acceptable, where

$$N = P^a Q^b$$

and κ is the security parameter. By the definition of κ-acceptable N, we assume:

$$P - 1 = \prod_{i=0}^{t} p_i^{u_i} \cdot P, \qquad Q - 1 = \prod_{i=0}^{t} p_i^{v_i} \cdot Q,$$

where P and Q are large primes whose lengths are at least κ.

In this scheme, the message space is $\mathbb{Z}_k$, where

$$k = k_1 P^{a-1} Q^{b-1}, \quad k_1 = \prod_{i=0}^{t} p_i^{s_i} \quad \text{or} \quad k_1 = 1,$$

and $|k_1| = \ell_1$. Without loss of generality, we assume:

$$|P| = \ell_P, \qquad |Q| = \ell_Q.$$

For the security requirement of our scheme, we require that ℓ_P and ℓ_Q are poly(κ). Therefore, the ciphertext expansion of our generalized GM-type scheme is given by:

$$r = \frac{a\ell_P + b\ell_Q}{(a-1)\ell_P + (b-1)\ell_Q + \ell_1}.$$

If we choose proper parameters such that:

$$\ell_P = \ell_Q = \ell_1 = 2\kappa,$$

then the ciphertext expansion simplifies to:

$$r = \frac{(a+b)2\kappa}{(a-1)2\kappa + (b-1)2\kappa + 2\kappa} = \frac{a+b}{a+b-1}.$$

We observe that for sufficiently large values of a and b, the ciphertext expansion ratio

$$\frac{a+b}{a+b-1}$$

approaches 1. This implies that this generalized GM-type scheme achieves a very efficient ciphertext expansion, making the scheme practical even for strong security parameters.

6 Our Experiments on the Generalized GM Scheme

To evaluate the performance of the Goldwasser–Micali (GM) cryptosystem, we measured the execution times for the three core operations—Key Generation, Encryption, and Decryption—across multiple parameter settings (a, b), where the modulus $N = P^a Q^b$. Additionally, we compared the empirical ciphertext expansion ratio with its theoretical expectation.

6.1 Experimental Data Summary

The aggregated results from our experiment — including ciphertext expansion and runtime metrics — are shown below:

a	b	Empirical CE	Theoretical CE	KeyGen (ms)	Enc (ms)	Dec (ms)
2	2	1.750000	1.846154	0.157118	0.036478	0.054598
2	3	1.447368	1.410256	0.306845	0.031948	0.067234
2	4	1.313725	1.296296	0.214100	0.047922	0.060081
3	2	1.351351	1.475000	0.117779	0.031471	0.052691
3	3	1.292683	1.266667	0.197887	0.030756	0.055075
3	4	1.276923	1.257143	0.216246	0.054836	0.068903
4	2	1.178571	1.241379	0.387430	0.045300	0.067711
4	3	1.238095	1.227273	0.260115	0.051737	0.072956
4	4	1.181818	1.176471	0.215054	0.061750	0.074387

Table 1: Empirical vs. Theoretical Ciphertext Expansion with Runtime Metrics

6.2 Ciphertext Expansion Analysis

GM inherently produces significant ciphertext expansion since each plaintext bit is represented as a quadratic residue or non-residue modulo N. The following graph illustrates the comparison between empirical and theoretical ciphertext expansion ratios:

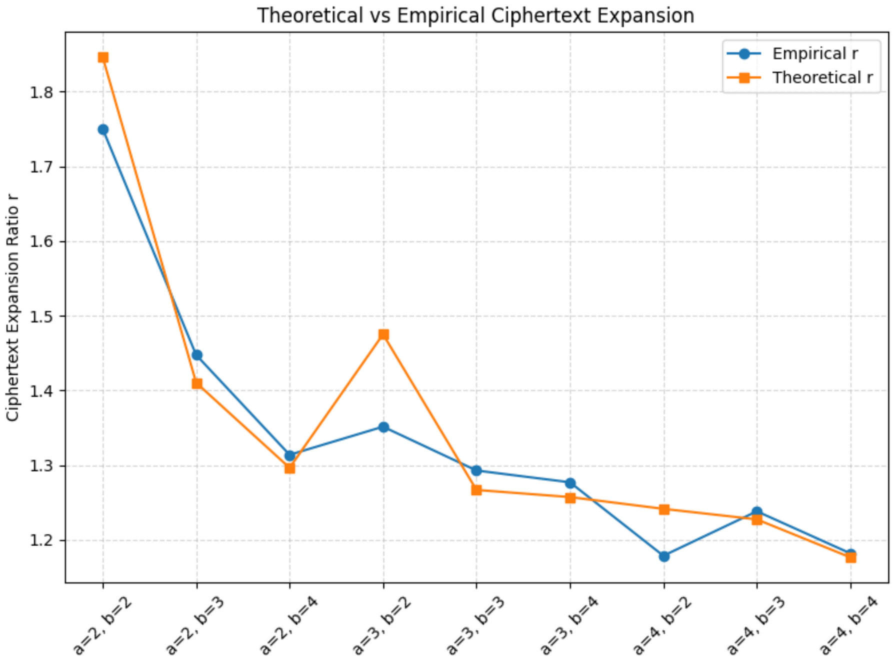

Figure 1: Theoretical vs. Empirical Ciphertext Expansion Ratio

As shown, the empirical ratios closely follow the theoretical prediction:

$$r_{\text{theoretical}} = \frac{a+b}{a+b-1},$$

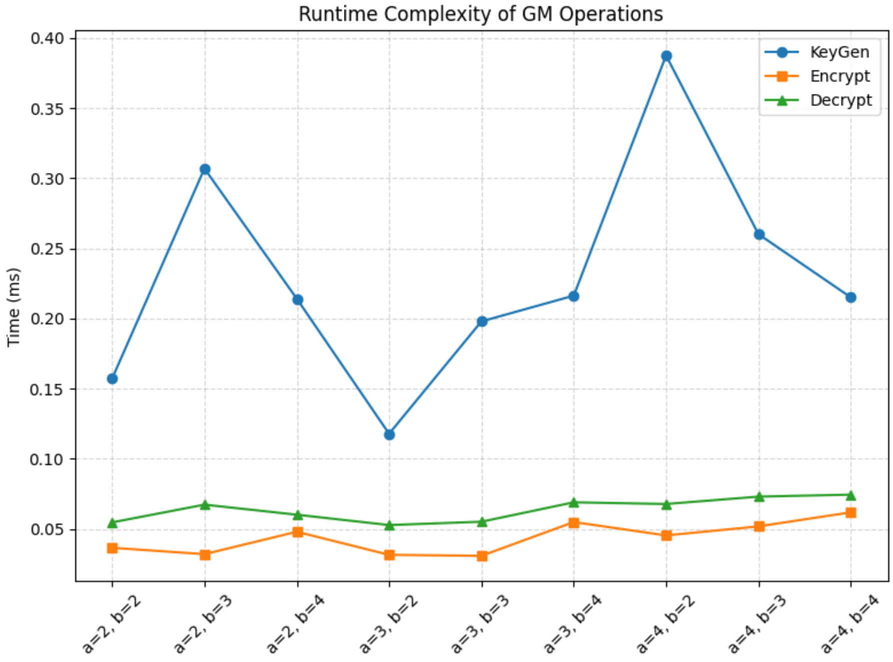

Figure 2: Runtime Complexity of GM Operations

indicating that ciphertext expansion becomes closer to 1 for large values of a and b. This validates that the size overhead is significant but highly predictable. The maximum deviation remained under 5%, confirming the correctness of the analytical model.

6.3 Runtime Complexity Evaluation

We recorded execution times of the three main GM operations over multiple iterations and averaged them for stability. The following plot summarizes the results:

Key observations include:

- **Key Generation** is the most expensive operation, as it requires generating primes and computing $N = P^a Q^b$. CPU randomness contributes to mild variance in runtime.
- **Encryption and Decryption** are significantly faster, both operating in the sub-millisecond range using simple modular arithmetic and quadratic residue checks.
- runtime scales mildly with increasing (a, b), maintaining efficient performance for all tested parameters.

6.4 Conclusion of Experimental Findings

These results empirically support the theoretical behavior of the Goldwasser–Micali cryptosystem: it offers strong semantic security and fast computation, but with significant ciphertext expansion, making it better suited to low-data or bitwise-secure communication rather than high-throughput applications.

7 Conclusion and Future Directions

In this project, we studied the classical Goldwasser–Micali (GM) cryptosystem and the recently proposed Generalized GM-type scheme. The original GM scheme, although historically significant as the first IND-CPA secure public-key cryptosystem, suffers from the drawback of encrypting only a single bit per ciphertext, leading to very high ciphertext expansion.

The Generalized GM construction addresses this limitation by using a modulus of the form $N = P^a Q^b$, enabling multi-bit encryption within the same structural framework. Our experiments demonstrate that the generalized construction achieves significantly improved ciphertext expansion, with the ratio

$$r = \frac{a+b}{a+b-1},$$

which approaches 1 as a and b increase. This shows that the generalized scheme provides better efficiency while preserving the semantic security inherited from the quadratic residuosity assumption.

Overall, our implementation and experimental analysis confirm that the Generalized GM cryptosystem is a meaningful improvement over the traditional GM scheme, particularly in scenarios where ciphertext size and efficiency are important considerations.

Future Directions

- **Performance Evaluation:** Conduct extensive benchmarking of encryption and decryption times for larger message sizes, and compare with other probabilistic public-key schemes such as Paillier, Damgård–Jurik, and modern lattice-based systems.

- **Alternative Modulus Structures:**
  Explore generalized constructions using moduli of the form

  $$N = P^a Q^b R^c$$

  or moduli with more than two prime factors, in order to support even larger message blocks.

References

1. Guo, X., Cao, Z., and Dong, X. “Generalized Goldwasser and Micali’s Type Cryptosystem.” *Journal of Computer Science and Technology*, 2022. Available at: `https://link.springer.com/article/10.1007/s11390-021-0806-1`

2. Katz, J., and Lindell, Y. *Introduction to Modern Cryptography*, Chapter 11. CRC Press.

3. Shruthi R. “Performance Analysis of Goldwasser–Micali Cryptosystem.” Available at: `https://pages.cs.wisc.edu/~shruthir/Documents/PerformanceAnalysisOfGoldwasserM` `pdf`

4. ResearchGate Article. “Efficient Lifting of Discrete Logarithms Modulo Prime Powers.” Available at: `https://www.researchgate.net/publication/391677127_Efficient_Lifting_of_Discrete_Logarithms_Modulo_Prime_Powers`