CS Gems Final

Tal Shachar , Gidon Abbas

March 2025

1 Background

1.1 Quadratic Residues

We say that a number $x \in \mathbb{N}$ is a Quadratic Residue (QR) $\mod N$, if there exists $y \in \mathbb{Z}_N$ s.t. $x \equiv y^2 \mod N$. A use we saw in class for Quadratic Residues is the Group Of Quadratic Residues modulus a prime number of the form $p = 2q+1$ where q is a prime. This is because exactly half of the non-zero elements in $\mathbb{Z}_p$ where p is an odd prime are QRs:

1.1.1 Amount of Quadratic Residues

Exactly as in the Problem Sheet we define a function $\varphi : \mathbb{Z}_p \to \mathbb{Z}_p$ as such: $\varphi(x) = x^2 \mod p$. Note that by definition, $img(\varphi)$ are all quadratic residues $\mod p$. Note that by field theorems $ker(\varphi) = \{0\}$ since there are no zero divisors in a field.

Let $y \in img(\varphi) \setminus \{0\}$. By definition there exists $x \in (\mathbb{Z}_p \setminus \{0\})$ s.t. $x^2 \equiv y$. Since p is an odd prime, $x \not\equiv -x \mod p$. Let's ATC that there exists $z \not\equiv \pm x \mod p$ s.t. $z^2 \equiv y \mod p$: then,

$$0 = \varphi(0) \not\equiv \varphi(z-x) = (z-x)^2 \mod p = z^2 - 2zx + x^2 \mod p = 2y - 2zx \mod p$$

Which gives us $2y \equiv 2zx \mod p \Rightarrow zx \equiv y \equiv x^2 \Rightarrow z \equiv x$. Contradiction. So overall we have exactly two sources for y in $\mathbb{Z}_p$, and φ is a 2-1 function (when ignoring zero). Since the image is all of the Quadratic Residues modulus p, we get that the $|QRs \mod p| = \frac{|\mathbb{Z}_p|}{2} = \frac{p-1}{2}$.

This conclusion is very important, because all of the logic below is based on the fact that the distinction between a non-zero quadratic residue and a non-zero non-quadratic residue is very difficult modulus a composite number. We will see some uses for this shortly.

1.1.2 Quadratic Residue Algebra

In order to prove some qualities about Quadratic Residues we will prove that the non-zero Quadratic Residues mod p form a subgroup of the multiplicative group $\mathbb{Z}_p^*$, which we will call QR_p^*. First note that $e_{\mathbb{Z}_p^*} = 1 \equiv 1^2 \in QR_p^*$. Let

$y_1, y_2 \in QR_p^*$. By definition there exist $x_1, x_2 \in \mathbb{Z}_p^*$ s.t. $x_1^2 \equiv y_1$ and $x_2^2 \equiv y_2$. $y_1 {y_2}^{-1} = x_1^2 (x_2^2)^{-1} = x_1^2 ({x_2}^{-1})^2 = (x_1 {x_2}^{-1})^2 \in QR_p^*$

This concludes the criteria for a subgroup.

Let $n_1, n_2, m_1, m_2 \in {\mathbb{Z}_{|}}^*$ s.t. $\{n_1, n_2, m_1, m_2\} \cap QR_p^* = \{n_1, n_2\}$. We will now show some properties:

1. $n_1 n_2 \in QR_p^*$: By group properties

2. $n_1 m_1 \notin QR_p^*$: By group properties, otherwise m_1 would be in QR_p^*

3. $m_1 m_2 \in QR_p^*$: Since $\mathbb{Z}_p^*$ is cyclic there exists a generator $g \in \mathbb{Z}_p$ and $s, r \in \mathbb{N}$ s.t. $m_1 \equiv g^s, m_2 \equiv g^r$. we know that $r \mod 2 = s \mod 2 = 1$ since m_1, m_2 are not quadratic residues. Therefore, $m_1 m_2 = g^s g^r = g^{s+r} = g^{2k}$ since $s + r \mod 2 = 0$. So, g^k is a root of $m_1 m_2$ and we have that $m_1 m_2$ is a quadratic residue.

1.2 Definitions

Given an odd prime p, we define the Legendre Symbol, which we will write as $Leg(x, p)$ to be a homomorphism from $\mathbb{Z}_p$ to the multiplicative group of order two containing 1 and -1. The homomorphism is defined as such:

$$Leg(x,p) = \begin{cases} 1 & \text{if } x \in QR_p^* \\ -1 & \text{if } x \notin QR_p^* \end{cases}$$

The proof for the homomorphic property follows immediately from the properties previously mentioned, since $\{-1, 1\}^* \cong \mathbb{Z}_2$.

We define an explicit mathematic formula formula for the Legendre symbol that we will prove soon to be $Leg(x, p) = x^{\frac{p-1}{2}} \mod p$.

1. If $x \in QR_p^*$, there exists $r \in \mathbb{Z}_p^*$ s.t. $x = r^2 \mod p$. Therefore $x^{\frac{p-1}{2}} \equiv (r^2)^{\frac{p-1}{2}} \equiv r^{p-1} \equiv 1$ by FLT.

2. We know that Leg is a a homomorphism, any function of the form $f(x) = x^n \mod p$ is a homomorphism (Seem in the problem sheets.) We see that the entirety of QR_p^* get sent to 1, so they are a subgroup of $ker(Leg)$. We also know that $|QR_p| = \frac{|\mathbb{Z}_p|}{2}$, so there cannot be any other element sent to 1 from $\mathbb{Z}_p$ otherwise it would contradict Lagrange's Theorem. So the only other possible image since $x^{\frac{p-1}{2}} = \sqrt{x^{p-1}} = \pm 1$, is -1.

We have now proved that the formula works. We will now define an extension of the Legendre Symbol, which is called the Jacobi symbol, that works beyond the confines of modulus prime numbers. Let $n = p_1 \cdot ... \cdot p_k$ where $p_1, ..., p_k$ are distinct odd primes. We define $Jac(x, n) = \Pi_{i=1}^k Leg(x, p_k)$ to be a function from $\mathbb{Z}_n^*$ to $\{-1, 1\}$, same as with the legendre symbol, but now the meaning is a bit different. We will see uses for this symbol in the Goldwasser-Micali crypto system.

2 Goldwasser-Micali's First Probabilistic Encryption Scheme

2.1 Introduction

The security of the Goldwasser-Micali encryption scheme is based on complexity theory, specifically the difficulty of extracting any meaningful information from a given ciphertext.

This scheme relies on the hardness of certain number-theoretic problems, such as factoring large integers and deciding quadratic residuosity modulo a composite number. To establish the security of our encryption scheme, we demonstrate that breaking it is at least as hard as solving these problems. That is, any attack on the encryption scheme would yield an efficient algorithm for deciding quadratic residuosity modulo composite integers.

2.2 Trapdoor Function Models

As discussed in class, a trapdoor function is a function $\mathcal{E}$ that is easy to compute but hard to invert. A classic example is the RSA function. However, RSA suffers from certain security limitations.

2.3 Introduction to Probabilistic Encryption and Modern Encryption

Traditional encryption schemes are deterministic functions, meaning that the same message always results in the same ciphertext. Probabilistic encryption introduces randomness to avoid this problem. This allows us to address security concerns associated with trapdoor functions while avoiding the need to impose any probability structure on the messages.

In this scheme, each message bit is encrypted independently, using a probabilistic approach. Given a message of length l, the encryption process requires k bits per message bit, where k is the bit length of N (where $N = pq$, with p, q being large odd primes).

2.3.1 Probabilistic Encryption Definition

Let $\mathcal{E}$ be an encryption algorithm that takes as input an l-bit binary message $m = m_1 \ldots m_l$. For each bit m_i, the algorithm randomly selects an element x and uses it to encrypt m_i, ensuring that a single plaintext message can have multiple possible ciphertext representations.

2.4 Implementation

Define the quadratic residuosity function:

$$Q_n(x) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue mod } n, \\ 0 & \text{otherwise.} \end{cases}$$

Given a probabilistic public key cryptosystem Π, the key generation works as follows:

1. Randomly select two k-bit primes p and q.
2. Compute $n = pq$.
3. Pick $y \in \mathbb{Z}_n^1 = \{x \in \mathbb{Z}_n^* : Jac(x,n) = 1\}$ such that y is a quadratic non-residue modulo n, meaning $Leg(x,p) = Leg(x,q) = -1$.
4. The public key is (n, y), and the private key is (p, q).

2.4.1 Encryption Procedure

Suppose Bob wants to send Alice a message $m = m_1, \ldots, m_l$. For each $m_i \in m$:

- Bob chooses a random number $x \in \mathbb{Z}_n^*$.
- If $m_i = 1$, then $e_i = yx^2 \mod n$.
- Otherwise, $e_i = x^2 \mod n$.

Bob sends the encrypted message $(e_1, \ldots, e_l) = \mathcal{E}(m)$. Encoding a l-bit message takes $O(lk^2)$ time, and each bit of plaintext expands into k bits of ciphertext.

2.4.2 Decryption Procedure

Alice receives the ciphertext $(e_1, \ldots, e_l)$. For each e_i:

- Alice computes $Q_n(e_i)$ using her private key (p, q).
- She reconstructs $m = m_1, \ldots, m_l$.

Recovering m from its encryption requires $O(lk^3)$ time.

2.4.3 Computing Quadratic Residuosity using the Jacobi Symbol

If Alice knows p and q, she can determine whether c is a quadratic residue modulo both p and q by computing:

- The Legendre symbol $(\frac{c}{p})$.
- The Legendre symbol $(\frac{c}{q})$.
- If both results are 1, then c is a quadratic residue, which means the original bit was 0.
- Otherwise, the original bit was 1.

2.5 Correctness Proof

To show that decryption correctly recovers the original message, we prove that $D(E(m)) = m$. Let $n = pq$, where p, q are large odd primes, and let $y \in \mathbb{Z}_n^1$ be a quadratic non-residue. Suppose $x \in \mathbb{Z}_n^*$.

- If $m = 0$, then:

$$D(E(m)) = D(x^2 \mod n) = \{(\frac{x^2}{p}), (\frac{x^2}{q})\} = \{1, 1\} \Rightarrow 0 = m.$$

- If $m = 1$, then:

$$D(E(m)) = D(yx^2 \mod n) = \{(\frac{yx^2}{p}), (\frac{yx^2}{q})\}.$$

  Since y is a quadratic non-residue, we get $\{-1, -1\}$, which implies $m = 1$.

2.6 Intuitive Proof of Complexity

As an informal proof of the difficulty of deciding whether a number $x \in \mathbb{Z}_n$ is a quadratic residue, we will show a reduction to the factorization problem. We assume that we have a deterministic oracle function R that given a number $x \in \mathbb{Z}_n$, gives us a square root of $x \mod n$, or *null* if no such root exists (i.e. x is not a QR).

Without knowing the factorization, we now show how to factorize n to p, q using R.

2.6.1 Root Count of QR mod n

Let $x \in \mathbb{Z}_n$ s.t. $Q_n(x) = 1$. Therefore, $Q_q(x) = Q_p(x) = 1$, meaninng that there exists roots $r_p \in \mathbb{Z}_p, r_q \in \mathbb{Z}_p$ s.t. $r_p^2 \equiv x \mod p$ and $r_q^2 \equiv x \mod q$. Because of the properties of square roots, we also get that $-r_p, -r_q$ are square roots of x mod p and q respectively. This gives us overall exactly two square roots for each prime divisor of n. Using the CRT in the same way we did in the 7'th problem sheet, we can take every one of the 4 combinations of roots from the two couples and find a number in $\mathbb{Z}_n$ that is a square root of $x \mod n$.

In conclusion, for every Quadratic Residue $x \in \mathbb{Z}_n$ there exist exactly 4 roots modulo n.

2.6.2 Factorization Algorithm

1. Generate a random number $x \in \mathbb{Z}_n$.
2. Take $r = R(x^2)$. Since x^2 is a square number, $r \neq nil$.
3. If $r = \pm x$, go back to step 1. Since there are exactly 4 square roots mod n for x, the probability for this happening is 0.5.

4. By definition $x^2 \equiv r^2 \mod n \Rightarrow x^2 - r^2 \equiv 0 \mod n$. Using an identity we see that $(x-r)(x+r) \equiv 0 \mod n$. Since $x \neq \pm r$, we know that both of the elements in the multiplication are not congruent to $0 \mod n$

From this point, we know that exactly one of the sides of the equation must have p as a divisor, and the other must have q as a divisor, since the multiplication is divisible by n. Using this we take $gcd(x-r, n) = p$ WLOG, and $q = \frac{n}{p}$. We have successfuly factored n into its prime number components, which is considered a hard problem, in a bounded polynomial time probabilistic algorithm. From this we conclude that the Quadratic Residuosity problem is at least as difficult as the factorization problem, and we have an intuition for why we would like to use it.

3 Paillier Cryptosystem

3.1 Introduction

Despite the fact that Goldwasser and Micali's encryption scheme revolutionized the world of cryptography with the introduction of the concept of probabilistic encryption schemes, it is not very practical in today's standard; a single bit of information sent is inflated to the amount of bits necessary to represent a value $x \in \mathbb{Z}_n$, which is an inflation ratio of $O(log(n))$. Since then, a lot of new encryption schemes have been invented, such as the El-Gamal encryption scheme studied in class, which offers a much better ratio and still produces a probabilistic scheme, while working in a very different way.

In order to expand upon the concept of residuality and the problems of deciding the residuality properties of a number modulus a composite number, we researched and found the Paillier Cryptosystem - a probabilistic homomorphic encryption that works under a different assumption but offers incredible innovative features

3.2 N-th residuosity problem

When we talked about the Goldwasser-Micali cryptosystem, the reduction used to show how the difficulty of deciding whether a number in $\mathbb{Z}_{pq}$ is a quadratic residue is the same as factoring pq into p, q.

In the Paillier Cryptosystem however, the assumption is different; given an $N = pq$ where p, q are odd prime numbers, the complexity assumption is that without knowing the factorization of n, deciding whether a number $x \in \mathbb{Z}_{N^2}$ is an N-th residue - meaning there exists $y \in \mathbb{Z}_{N^2}$ s.t. $x \equiv y^N$ - is a hard problem.

3.3 Setup

The Paillier Encryption Scheme is an asymmetric encryption scheme. Given two odd prime numbers p, q, the public key is only $N = pq$, and the private key is the factorization of N, p, q. We also choose a number $g \in \mathbb{Z}_{N^2}$ of order

N, commonly chosen as $g = (1 + N) \mod N^2$, as according to the binomial formula all values except for 1 become divisible by N^2 and thus cancel out, only when reaching the power of N.

3.4 Implementation

3.4.1 Encryption Procedure

In order to encrypt a message $m \in \mathbb{Z}_n$, we generate a uniform element $r \in \mathbb{Z}^*_{N^2}$. The ciphertext is then defined as $c = g^m r^N \mod N^2 = (1+N)^m \cdot r^N \mod N^2$. We won't go in too much depth into why this is done, but the simple idea is as such; an expansion of deciding whether a number is an N-th residue, is deriving the so-called residuosity-class of a value $x \in \mathbb{Z}_{N^2}$, which is the distinct $y \in \mathbb{Z}_N$ for which there exists $r \in \mathbb{Z}_N$ such that $x = g^y \cdot r^N$. Essentially it comes to separating the power of the special number g we chose earlier. That Residuosity class, often notated $\omega_g(x)$, is precisely the discrete value used to encode the encrypted message.

3.4.2 Decryption Procedure

We define $L(x) : \mathbb{Z}_{N^2} \to \mathbb{Z}_N$ to be a function that finds $l \in \mathbb{Z}_N$ s.t. lN is the biggest multiple of N that is smaller than $x - 1$, and is defined with respect to normal real number division and size relation and not modulus: $L(x) = \lfloor \frac{x-1}{N} \rfloor$.

We then finally compute the message given a ciphertext $c \in \mathbb{Z}_{N^2}$ to be

$$m = \frac{L(c^{\varphi(N)} \mod N^2)}{\varphi(N) \mod N}$$

This is one monster of an expression, but the idea is really simple - we want to 'get rid' of the r-part of the value defined as c and get only the exponent of g. To do that we need to set some ground work:

3.4.3 Order of $\mathbb{Z}^*_{N^2}$

To find the order of $\mathbb{Z}^*_{N^2}$, we must identify all non-divisble elements. Since $N^2 = N \cdot N = pq \cdot pq = p^2q^2$, we have that for $x \in \mathbb{Z}_{N^2}$ $gcd(x, N^2) = 1 \iff gcd(x, p) = gcd(x, q) = 1$ We find that in each partition of size N of $\mathbb{Z}^*_{N^2}$ the amount of elements that satisfy the right side are the same and there are $\varphi(N)$ elements as such, so overall we have $|\mathbb{Z}^*_{N^2}| = \varphi(N) \cdot N$

3.4.4 Back to Correctness

After establishing the size of the Group $\mathbb{Z}^*_{N^2}$, we can prove the correctness of the equality in the decryption procedure:

$$\begin{aligned} c^{\varphi(N)} &= (g^m \cdot r^N)^{\varphi(N)} = g^{\varphi(N)\cdot m} \cdot r^{\varphi(N)\cdot N} && \text{Raising element to the order of the group } \varphi(N)\cdot N \\ &= g^{\varphi(N)\cdot m} = (1+N)^{\varphi(N)\cdot m} && \text{Binomial formula w.r.t } N^2 \equiv 0 \\ &= 1 + \varphi(N)\cdot m \cdot N \end{aligned}$$

Now we can calculate $L(c^{\varphi(N)})$:

$$\begin{aligned} L(c^{\varphi(N)}) &= L(1+\varphi(N)\cdot m\cdot N) \\ &= \lfloor \frac{(1+\varphi(N)\cdot m \cdot N)-1}{N} \rfloor && \text{Definition of } L \\ &= \lfloor \frac{\varphi(N)\cdot m\cdot N}{N} \rfloor = \lfloor \varphi(N)\cdot m \rfloor = \varphi(N)\cdot m \end{aligned}$$

Putting everything together, we get

$$\frac{L(c^{\varphi(N)} \mod N^2)}{\varphi(N) \mod N} = \frac{\varphi(N)\cdot m \mod N}{\varphi(N) \mod N} = m \mod N$$

Precisely as required.

Note Note that in order to cancel out the r part of the ciphertext, we had to know $\varphi(N)$, which by the RSA assumption is a hard problem that reduces down to factorization of pq.

3.5 Homomorphic Encryption Property

A very interesting property that arises from the definition of the Paillier Encryption is as follows: for two messages $m_1, m_2 \in \mathbb{Z}_N$, $Dec(Enc(m_1)\cdot Enc(m_2)) = Dec((g^{m_1}r_1^N)\cdot(g^{m_2}r_2^N)) = Dec(g^{m_1+m_2}r^{2N})$. Continuing the decryption logic as in the correctness proof we finally get the result that $Dec(Enc(m_1)\cdot Enc(m_2)) \equiv_M m_1 + m_2$.

This allows the party that does not have the private to do additive operations on sets of encrypted values, without knowing the data inside. This homomorphic property appears in many other encryption algorithms, but not in the same way shown here

1. Goldwasser-Micali: In this cryptosystem, given that the encryption is of a single bit, we also have an "additive" homomorphism; but since the field we are working under is $\mathbb{Z}_2$, we effectively only get a xor on the messages sent, since we split each message to its bits and encrypt seperately.

2. El-Gamal: For El-Gamal we have a multiplicative homomorphism - $Dec(Enc(m_1)\cdot Enc(m_2)) = m_1m_2$which can also be very important, for example for detecting whether zero appears in a set of numbers, without knowing anything about any singular number.

3.6 Example of A "Secure" Rating System

Using the additive homomorphic property we derived, and assuming that we can use some sort of a Zero-Knowledge-Range-Proof (Which we currently do not have the mathematical tools to understand but we know exist), we can allow each party to give a score from 1-to-s by sending $r_1, ..., r_s$ different messages, constrained under a zero knowledge proof to fulfill $Dec(r_i) \in \{0 \mod N, 1 \mod N\}$. Each party will send $r_1, ..., r_s$ to a middle party that does not have the secret key, which will verify the Zero-Knowledge-Range-Proof as mentioned. For each party $1 \leq j \leq m$ we define s_j to be $\Pi_{i=1}^{s} r_i$ with the respective $r_1, ..., r_s$ for that party. Finally, the middle party will send $\Pi_{j=1}^{m} s_j$ to the private-key owner, which will decrypt the value to be $Dec(\Pi_{j=1}^{m} s_j) = \Sigma_{j=1}^{m} Dec(s_j) = \Sigma_{j=1}^{m} \Sigma_{i=1}^{s} Dec(r_i) = \text{total score} = S$. Finally, by taking $\frac{S}{m}$ when treating S as a whole number, we get the average rating of all users, without any party at any point being able to distinguish between different scores given by any person (all parties might have given the same score, the distribution may vary while unbeknownst to the private-key owner). We personally think this is a really elegant and interesting use of this property, and are really glad we got to come across it.