Lattice Based Cryptography and Fully Homomorphic Encryption

Ani Nadiga

Carleton College

NUMS
Introduction to Cryptography

The most basic encryption scheme you can think of - Caesar Cipher
Introduction to Cryptography

The most basic encryption scheme you can think of - Caesar Cipher

![Caesar Cipher Diagram](https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex)

**Figure 1:** [https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex](https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex)
Introduction to Cryptography

The most basic encryption scheme you can think of - Caesar Cipher

Figure 1: https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex

This scheme is super easy to break, so we needed something more
Public Key Cryptosystem

Alice

Secret Key
Public Key Cryptosystem

Alice

Secret Key

Public Key
Public Key Cryptosystem

Alice

Secret Key

Public Key

Bob
Public Key Cryptosystem

Alice

Secret Key

Public Key

Bob

$m = message$
Public Key Cryptosystem

Alice

Secret Key

Public Key

Enc(m) → Public Key → m

Bob

m = message
Public Key Cryptosystem
Public Key Cryptosystem

Alice

Secret Key

Public Key

Enc(m) → Public Key  m

Eve

Bob

m = message
Public Key Cryptosystem

Alice

Secret Key

m ← Private Key → Enc(m)

Public Key

Enc(m) ← Public Key → m

m = message

Bob
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

Given the public key it is hard to find the private key because factoring large integers is hard.

RSA is based on the integer factoring problem being hard.

But with the private key it is easy!
RSA

Secret Key - two large prime numbers

With just the public key, finding $m$ given Enc($m$) is hard, but with the private key it is easy!

Given the public key it is hard to find the private key because factoring large integers is hard.

RSA is based on the integer factoring problem being hard.
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

\[
\text{Public Key} \\
m \rightarrow \text{Enc}(m)
\]
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

\[
\text{Public Key} \quad m \quad \text{Enc}(m)
\]

With just the public key, finding \( m \) given \( \text{Enc}(m) \) is hard,
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

\[
\text{Public Key} \quad m \quad \rightarrow \quad \text{Enc}(m)
\]

With just the public key, finding \( m \) given \( \text{Enc}(m) \) is hard,
But with the private key it is easy!
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

\[
\begin{align*}
\text{Public Key} \\
m \quad \quad \rightarrow \quad \quad \text{Enc}(m)
\end{align*}
\]

With just the public key, finding \( m \) given \( \text{Enc}(m) \) is hard,
But with the private key it is easy!

Given the public key it is hard to find the private key because factoring large integers is hard
RSA

Secret Key - two large prime numbers
Public Key - product of those prime numbers

\[ m \rightarrow \text{Public Key} \rightarrow \text{Enc}(m) \]

With just the public key, finding \( m \) given \( \text{Enc}(m) \) is hard,
But with the private key it is easy!

Given the public key it is hard to find the private key because factoring
large integers is hard
RSA is based on the integer factoring problem being hard
Short Comings of RSA

Quantum algorithms can factor integers efficiently

▶ Quantum computers can break all our cryptography!

Not provably secure

▶ For some choices of primes RSA can be broken without factoring the public key

Can not process on encrypted data

▶ Given Enc(a) and Enc(b), can not find Enc(a + b) or Enc(a · b)
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently

Not provably secure

For some choices of primes RSA can be broken with out factoring the public key

Can not process on encrypted data

Given Enc(a) and Enc(b), can not find Enc(a + b) or Enc(a · b)
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   - Quantum computers can break all our cryptography!

Ani Nadiga (Carleton College)
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   ▶ Quantum computers can break all our cryptography!
2. Not provably secure
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   - Quantum computers can break all our cryptography!

2. Not provably secure
   - For some choices of primes RSA can be broken without factoring the public key
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   - Quantum computers can break all our cryptography!

2. Not provably secure
   - For some choices of primes RSA can be broken without factoring the public key

3. Can not process on encrypted data
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   - Quantum computers can break all our cryptography!

2. Not provably secure
   - For some choices of primes RSA can be broken without factoring the public key

3. Can not process on encrypted data
   - Given $\text{Enc}(a)$ and $\text{Enc}(b)$, can not find $\text{Enc}(a + b)$ or $\text{Enc}(a \cdot b)$
Short Comings of RSA

1. Quantum algorithms can factor integers efficiently
   ▶ Quantum computers can break all our cryptography!

2. Not provably secure
   ▶ For some choices of primes RSA can be broken without factoring the public key

3. Can not process on encrypted data
   ▶ Given Enc(a) and Enc(b), can not find Enc(a + b) or Enc(a \cdot b)
Building a Better System
Building a Better System

We need a new problem to build a new crypto system on
Building a Better System

We need a new problem to build a new crypto system on

\[
\begin{array}{c}
25 \\
105 \\
35 \\
75 \\
15 \\
10 \\
\end{array}
\]
Building a Better System

We need a new problem to build a new crypto system on

\[
\begin{array}{ccc}
25 & & 36 \\
105 & & 100 \\
35 & & 24 \\
75 & & 84 \\
15 & & 65 \\
10 & & 4 \\
\end{array}
\]
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$.
Pick one $s \in \mathbb{Z}_q^n$.
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$
Pick one $s \in \mathbb{Z}_q^n$
Pick many $a_i \in \mathbb{Z}_q^n$
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$
Pick one $s \in \mathbb{Z}_q^n$
Pick many $a_i \in \mathbb{Z}_q^n$

Given $(a_1, a_1 \cdot s)$, $(a_2, a_2 \cdot s)$, $(a_3, a_3 \cdot s)$, can you find $s$?

Ani Nadiga (Carleton College)
Lattice Based Cryptography
NUMS 7 / 21
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$
Pick one $s \in \mathbb{Z}_q^n$
Pick many $a_i \in \mathbb{Z}_q^n$

Given $(a_1, a_1 \cdot s)$, $(a_2, a_2 \cdot s)$, $(a_3, a_3 \cdot s)$, ... can you find $s$?

$\chi$ an error distribution over $\mathbb{Z}_q^n$
Pick many $e_i \leftarrow \chi$
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$

Pick one $s \in \mathbb{Z}_q^n$

Pick many $a_i \in \mathbb{Z}_q^n$

$\chi$ an error distribution over $\mathbb{Z}_q^n$

Pick many $e_i \leftarrow \chi$

Set $b_i = a_i \cdot s + e_i$

Given $(a_1, a_1 \cdot s)$

Given $(a_2, a_2 \cdot s)$

Given $(a_3, a_3 \cdot s)$

... can you find $s$?
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$

Pick one $s \in \mathbb{Z}_q^n$

Pick many $a_i \in \mathbb{Z}_q^n$

Given $(a_1, a_1 \cdot s)$

Given $(a_2, a_2 \cdot s)$ can you find $s$?

Given $(a_3, a_3 \cdot s)$

...

$\chi$ an error distribution over $\mathbb{Z}_q^n$

Pick many $e_i \leftarrow \chi$

Set $b_i = a_i \cdot s + e_i$

Given $(a_1, b_1)$

Given $(a_2, b_2)$, finding $s$ is hard!

Given $(a_3, b_3)$

...

Ani Nadiga (Carleton College)
The Learning With Errors Problem

We work in $\mathbb{Z}_q^n$
Pick one $s \in \mathbb{Z}_q^n$
Pick many $a_i \in \mathbb{Z}_q^n$

$\chi$ an error distribution over $\mathbb{Z}_q^n$
Pick many $e_i \leftarrow \chi$
Set $b_i = a_i \cdot s + e_i$

Given $(a_1, a_1 \cdot s)$
$(a_2, a_2 \cdot s)$
$(a_3, a_3 \cdot s)$
... can you find $s$?

Given $(b_1, a_1)$
$(b_2, a_2)$
$(b_3, a_3)$
... finding $s$ is hard!

By adding a small amount of error a trivial problem becomes hard
Basic Scheme [BGV12]

Use the ring $R_q = \mathbb{Z}_q[x]/\langle x^d + 1 \rangle$

$\chi$ is the error distribution (over $R_q$)

$N = \lceil \log q \rceil$ number of samples for dRLWE to be well defined

Secret Key Generation:

pick $s' \leftarrow R_q,$

set SK: $s = (1, s') \in R_q^2$

Public Key Generation:

pick $a' \leftarrow R_q^N$ and $R_q^N \ni e \leftarrow \chi^N$

$b \leftarrow a's' + 2e.$

set PK: $A = \begin{bmatrix} b & -a' \end{bmatrix} \in R_q^{N \times 2}$

Note that $A \cdot s = 2e \in R_q^N$
Basic Scheme Cont.

Encryption:
message $m \in R_2$, $m = (m, 0) \in R^2_q$
$r \leftarrow R^N_2$ a small random vector
ciphertext $c = m + A^T r = \begin{bmatrix} m \\ 0 \end{bmatrix} + \begin{bmatrix} b^T r \\ -a'^T r \end{bmatrix} \in R^2_q$

Decryption:
for a ciphertext $c$ output $m \leftarrow \left[\left[\langle c, s \rangle \right]_q\right]_2$

$\langle c, s \rangle = \left\langle \begin{bmatrix} (a'^T s' + 2e^T) r + m \\ -a'^T r \end{bmatrix}, \begin{bmatrix} 1 \\ s' \end{bmatrix} \right\rangle = 2e^T r + m$

As long as $\langle c, s \rangle < q/2$ then $\left[\left[\langle c, s \rangle \right]_q\right]_2 = [2e^T r + m]_2 = m$

$[x]_q$ denotes taking an $0 \leq x \leq q - 1$ to its representative in $(-q/2, q/2]$
Addition and Multiplication

For two ciphertexts $c_1, c_2$ encrypting messages $m_1, m_2$

**Addition:** $c_1 + c_2$ represents $m_1 + m_2$

$$c_1 + c_2 = \begin{bmatrix} m_1 + b^T r_1 \\ -a'^T r_1 \end{bmatrix} + \begin{bmatrix} m_2 + b^T r_2 \\ -a'^T r_2 \end{bmatrix} = \begin{bmatrix} m_2 + m_1 + b^T (r_1 + r_2) \\ -a'^T (r_1 + r_2) \end{bmatrix}$$

$$\langle (c_1 + c_2), s \rangle = 2e^T (r_1 + r_2)$$

**Multiplication:** $c_1 \otimes c_2$ encrypts $m_1 \cdot m_2$ under the new key $s \otimes s$

$$m_1 \cdot m_2 = \left[ \left[ \langle c_1 \otimes c_2, s \otimes s \rangle \right]_q \right]_2$$
Recall that we are trying to build a crypto system that is:

1. Immune to quantum attacks
2. Provably secure
3. Capable of processing encrypted data
Recall that we are trying to build a crypto system that is:

1. **Immune to quantum attacks**
2. **Provably secure**
3. **Capable of processing encrypted data**

Also, how do we show that LWE problem is hard?
Lattice Problems

What is a lattice?
- A discrete additive subgroup of $\mathbb{R}^n$
- All linear combinations of some basis vectors

Lattices can exist in any dimension

Lattice Problems:
- Shortest Vector Problem
- Closest Vector Problem

These problems are conjectured to be both classically and quantum hard
Lattice Problems

What is a lattice?
- A discrete additive subgroup of $\mathbb{R}^n$
- All linear combinations of some basis vectors

Lattices can exist in any dimension

Lattice Problems:
- Shortest Vector Problem
- Closest Vector Problem

These problems are conjectured to be both classically and quantum hard
Lattice Problems

What is a lattice?

- A discrete additive subgroup of $\mathbb{R}^n$
- All linear combinations of some basis vectors

Lattices can exist in any dimension

Lattice Problems:

- Shortest Vector Problem
- Closest Vector Problem

These problems are conjectured to be both classically and quantum hard
The SVP LWE Reduction

How does this make LWE quantum hard?
The SVP LWE Reduction

How does this make LWE quantum hard?

Reduction

If there is a reduction from a problem A to a problem B, then an efficient algorithm for solving B can be used as a subroutine to make an efficient algorithm to solve problem A.
The SVP LWE Reduction

How does this make LWE quantum hard?

Reduction

If there is a reduction from a problem A to a problem B, then an efficient algorithm for solving B can be used as a subroutine to make an efficient algorithm to solve problem A

[Regev 05] found a quantum reduction from LWE to SVP
If you can solve LWE efficiently, then you can solve SVP efficiently
The SVP LWE Reduction

How does this make LWE quantum hard?

**Reduction**

If there is a reduction from a problem A to a problem B, then an efficient algorithm for solving B can be used as a subroutine to make an efficient algorithm to solve problem A.

[Regev 05] found a quantum reduction from LWE to SVP. If you can solve LWE efficiently, then you can solve SVP efficiently.

The encryption is an instance of LWE, so we have provable security.
The SVP LWE Reduction

How does this make LWE quantum hard?

**Reduction**

If there is a reduction from a problem A to a problem B, then an efficient algorithm for solving B can be used as a subroutine to make an efficient algorithm to solve problem A.

[Regev 05] found a quantum reduction from LWE to SVP.
If you can solve LWE efficiently, then you can solve SVP efficiently.

The encryption is an instance of LWE, so we have provable security.

We also have average case worst case reductions.
Recall that we are trying to build a crypto system that is:

1. Immune to quantum attacks
2. Provably secure
3. Capable of processing encrypted data
Recall that we are trying to build a crypto system that is:

1. Immune to quantum attacks
2. Provably secure
3. Capable of processing encrypted data
Homomorphic Encryption

Homomorphic Encryption

A form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. - Wikipedia
Homomorphic Encryption

Homomorphic Encryption

a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. - Wikipedia

Recall: given \( \text{Enc}(a) \) and \( \text{Enc}(b) \) we want \( \text{Enc}(a + b) \) and \( \text{Enc}(a \cdot b) \)
Homomorphic Encryption

Homomorphic Encryption

a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. - Wikipedia

Recall: given Enc(a) and Enc(b) we want Enc(a + b) and Enc(a \cdot b)

Homomorphic Encryption does not exist with traditional crypto tools
Homomorphic Encryption

a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. - Wikipedia

Recall: given Enc\((a)\) and Enc\((b)\) we want Enc\((a + b)\) and Enc\((a \cdot b)\)

Homomorphic Encryption does not exist with traditional crypto tools

In 2009, the first HE scheme was developed [Gentry 09], but was very slow
Homomorphic Encryption

a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. - Wikipedia

Recall: given Enc(a) and Enc(b) we want Enc(a + b) and Enc(a \cdot b)

Homomorphic Encryption does not exist with traditional crypto tools

In 2009, the first HE scheme was developed [Gentry 09], but was very slow

In 2013 a faster scheme was developed
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:

\[
m \xrightarrow{\text{RSA}} c
\]

This prevents "observational attacks"
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:

\[ m \xrightarrow{\text{RSA}} c \]
\[ m \xrightarrow{\text{RSA}} c \]

This prevents "observational attacks"
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:

\[ m \xrightarrow{\text{RSA}} c \]
\[ m \xrightarrow{\text{RSA}} c \]
\[ m \xrightarrow{\text{LC}} c_1 + e_1 \]

This prevents ”observational attacks”
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:

\[ m \xrightarrow{RSA} c \]
\[ m \xrightarrow{RSA} c \]
\[ m \xrightarrow{LC} c_1 + e_1 \]
\[ m \xrightarrow{LC} c_1 + e_2 \]
Why it Works

There are many aspects of the LWE problem that make homomorphic encryption possible, but one of the most important is that there is some randomness in the encryption:

\[
\begin{align*}
    m \xrightarrow{\text{RSA}} c \\
    m \xrightarrow{\text{RSA}} c \\
    m \xrightarrow{\text{LC}} c_1 + e_1 \\
    m \xrightarrow{\text{LC}} c_1 + e_2
\end{align*}
\]

This prevents "observational attacks"
Recall that we are trying to build a crypto system that is:

1. Immune to quantum attacks
2. Provably secure
3. Capable of processing encrypted data
Recall that we are trying to build a crypto system that is:

1. Immune to quantum attacks
2. Provably secure
3. Capable of processing encrypted data
What I did
What I did

Goal: get information from node A to node B, transmission line is untrusted
What I did

Goal: get information from node A to node B, transmission line is untrusted

So we add relay stations
What I did

Goal: get information from node A to node B, transmission line is untrusted

But information quality can degrade over long transmission lines
What I did

Goal: get information from node A to node B, transmission line is untrusted

So we add "relay stations"
Problems and Solutions

How do relay stations know what is degradation and what is the valid encryption with out knowing the unencrypted message?

Using homomorphic encryption techniques, we can check that transmitted information is correct with out knowing the message. But homomorphic evaluation causes the encryption's "noise" to grow, which increases the chances of decryption error.

We applied existing "noise management" techniques that do not compromise security when adding information that did not need to be encrypted, we found a way to incorporate unencrypted information with the encrypted information.
Problems and Solutions

How do relay stations know what is degradation and what is the valid encryption with out knowing the unencrypted message?

- Using homomorphic encryption techniques, we can check that transmitted information is correct with out knowing the message.
How do relay stations know what is degradation and what is the valid encryption without knowing the unencrypted message?

- Using homomorphic encryption techniques, we can check that transmitted information is correct without knowing the message.

But homomorphic evaluation causes the encryption’s “noise” to grow, which increases the chances of decryption error.
Problems and Solutions

How do relay stations know what is degradation and what is the valid encryption with out knowing the unencrypted message?

- Using homomorphic encryption techniques, we can check that transmitted information is correct with out knowing the message.

But homomorphic evaluation causes the encryption’s ”noise” to grow, which increases the chances of decryption error.

- We applied existing ”noise management” techniques that do not compromise security
Problems and Solutions

How do relay stations know what is degradation and what is the valid encryption without knowing the unencrypted message?

- Using homomorphic encryption techniques, we can check that transmitted information is correct without knowing the message.

But homomorphic evaluation causes the encryption’s ”noise” to grow, which increases the chances of decryption error.

- We applied existing ”noise management” techniques that do not compromise security
- When adding information that did not need to be encrypted, we found a way to incorporate unencrypted information with the encrypted information
1. **Regular LWE:**
   

2. **RLWE:**
   
Fully Homomorphic Encryption Schemes

1. Initial scheme by Gentry. Based on ideal lattices and uses the bootstrapping technique.
   

2. **RLWE Schemes:**
   
   1. **FHE without bootstrapping:**
      
   
   2. **FHE Batching:**
      