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

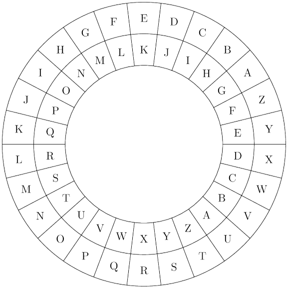


Figure 1: https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex

Introduction to Cryptography

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

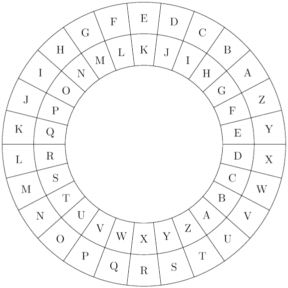
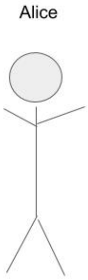


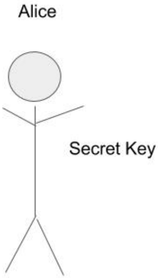
Figure 1: https://tex.stackexchange.com/questions/103364/how-to-create-a-caesars-encryption-disk-using-latex

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

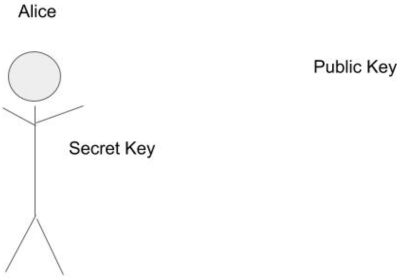
Public Key Cryptosystem



Public Key Cryptosystem



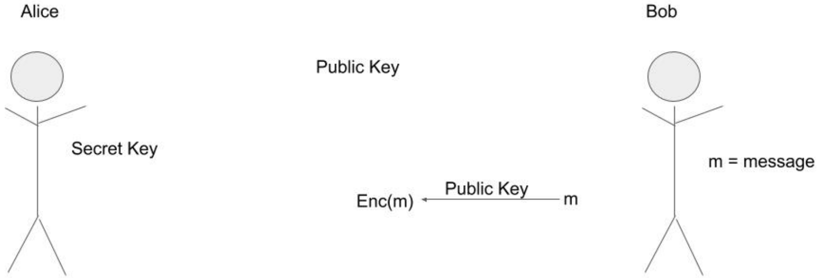
Public Key Cryptosystem



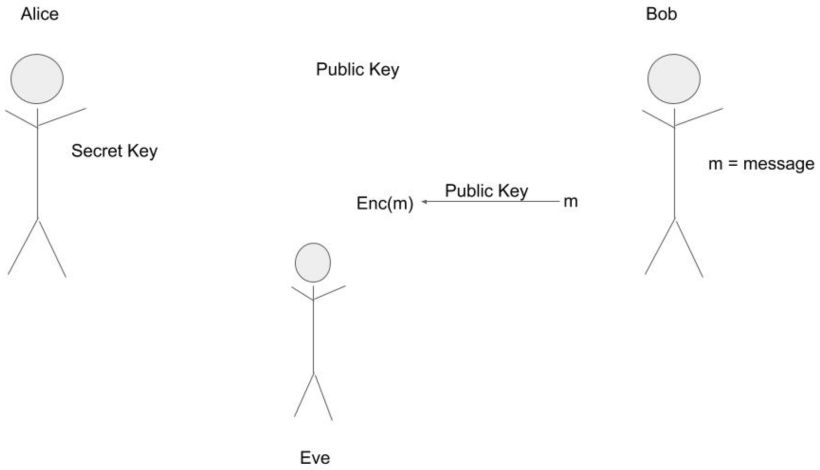
Public Key Cryptosystem

Public Key Cryptosystem

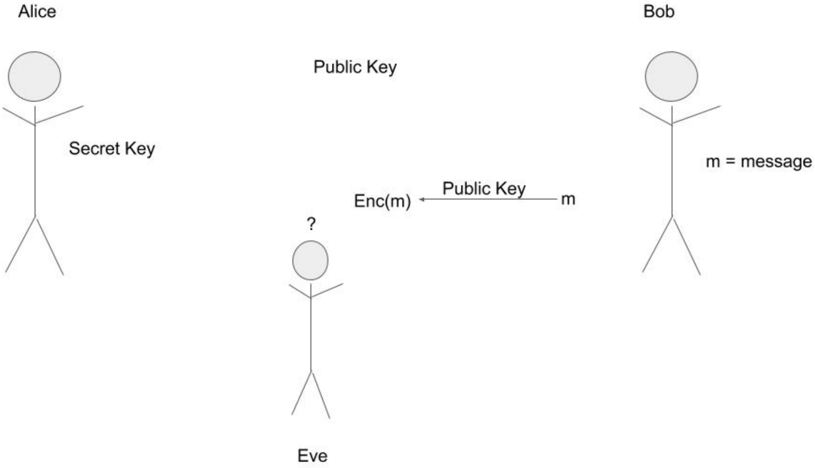
Public Key Cryptosystem



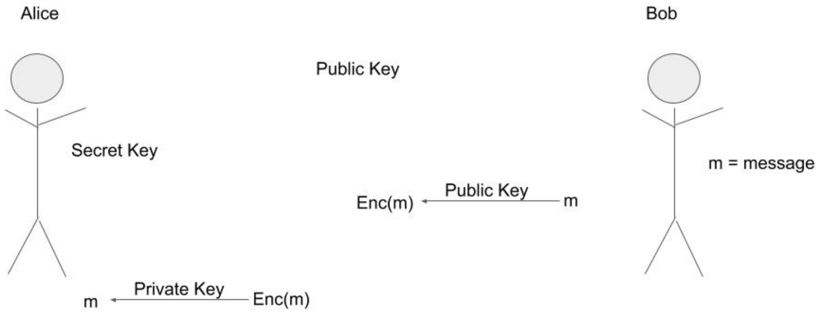
Public Key Cryptosystem



Public Key Cryptosystem



Public Key Cryptosystem



RSA

RSA

Secret Key - two large prime numbers

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

$$\text{m} \xrightarrow{\text{Public Key}} \text{Enc(m)}$$

RSA

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

$$m \xrightarrow{\text{Public Key}} \text{Enc}(m)$$

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

$$\text{m} \xrightarrow{\text{Public Key}} \text{Enc(m)}$$

With just the public key, finding m given 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

$$\text{m} \xrightarrow{\text{Public Key}} \text{Enc(m)}$$

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

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

$$\text{m} \xrightarrow{\text{Public Key}} \text{Enc(m)}$$

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

Short Comings of RSA

Short Comings of RSA

1. Quantum algorithms can factor integers efficiently

Short Comings of RSA

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

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 with out 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 with out 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 with out 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($a \cdot b$)

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 with out factoring the public key
3. Can not process on encrypted data
   - Given $\mathrm{Enc}(a)$ and $\mathrm{Enc}(b)$, can not find $\mathrm{Enc}(a + b)$ or $\mathrm{Enc}(a \cdot b)$

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

25
105
35
75
15
10

Building a Better System

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

25	36
105	100
35	24
75	84
15	65
10	4

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 $\begin{matrix}(a_1, a_1 \cdot s)\\(a_2, a_2 \cdot s)\\(a_3, a_3 \cdot s)\\ \dots\end{matrix}$ 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 $a_i \in \mathbb{Z}_q^n$

χ an error distribution over $\mathbb{Z}_q^n$
Pick many $e_i \leftarrow \chi$

Given $\begin{smallmatrix} (a_1, a_1 \cdot s) \\ (a_2, a_2 \cdot s) \\ (a_3, a_3 \cdot s) \\ \dots \end{smallmatrix}$ 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 $a_i \in \mathbb{Z}_q^n$

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

Given $\begin{matrix} (a_1, a_1 \cdot s) \\ (a_2, a_2 \cdot s) \\ (a_3, a_3 \cdot s) \\ \dots \end{matrix}$ 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 $\begin{matrix}(a_1, a_1 \cdot s)\\(a_2, a_2 \cdot s)\\(a_3, a_3 \cdot s)\\\cdots\end{matrix}$ can you find s?

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

Given $\begin{matrix}(a_1, b_1)\\(a_2, b_2)\\(a_3, b_3)\\\cdots\end{matrix}$, finding s is hard!

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 $\begin{matrix}(a_1, a_1 \cdot s) \\ (a_2, a_2 \cdot s) \\ (a_3, a_3 \cdot s) \\ \cdots\end{matrix}$ can you find s?

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

Given $\begin{matrix}(a_1, b_1) \\ (a_2, b_2) \\ (a_3, b_3) \\ \cdots\end{matrix}$, finding s is hard!

By adding a small amount of error a trivial problem becomes hard

Basic Scheme [BGV12]

Use the ring $R_q = \mathbb{Z}_q[x]/\langle x^d + 1 \rangle$
χ is the error distribution (over R_q)
$N = \lfloor \log q \rfloor$ number of samples for dRLWE to be well defined

Secret Key Generation:

pick $s' \leftarrow R_q$,
set SK: $\mathbf{s} = (1, s') \in R_q^2$

Public Key Generation:

pick $\mathbf{a}' \leftarrow R_q^N$ and $R_q^N \ni \mathbf{e} \leftarrow \chi^N$
$\mathbf{b} \leftarrow \mathbf{a}'s' + 2\mathbf{e}$.

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

Note that $\mathbf{A} \cdot \mathbf{s} = 2\mathbf{e} \in R_q^N$

Basic Scheme Cont.

Encryption:

message $m \in R_2$, $\mathbf{m} = (m, 0) \in R_q^2$

$\mathbf{r} \leftarrow R_2^N$ a small random vector

ciphertext $\mathbf{c} = \mathbf{m} + \mathbf{A}^T\mathbf{r} = \begin{bmatrix} m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}^T\mathbf{r} \\ -\mathbf{a'}^T\mathbf{r} \end{bmatrix} \in R_q^2$

Decryption:

for a ciphertext $\mathbf{c}$ output $m \leftarrow [[\langle \mathbf{c}, \mathbf{s} \rangle]_q]_2$

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

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

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

Addition and Multiplication

For two ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ encrypting messages m_1, m_2

Addition: $\mathbf{c}_1 + \mathbf{c}_2$ represents $m_1 + m_2$

$$\mathbf{c}_1 + \mathbf{c}_2 = \begin{bmatrix} m_1 + \mathbf{b}^T\mathbf{r_1} \\ -\mathbf{a'}^T\mathbf{r_1} \end{bmatrix} + \begin{bmatrix} m_2 + \mathbf{b}^T\mathbf{r_2} \\ -\mathbf{a'}^T\mathbf{r_2} \end{bmatrix} = \begin{bmatrix} m_2 + m_1 + \mathbf{b}^T(\mathbf{r_1} + \mathbf{r_2}) \\ -\mathbf{a'}^T(\mathbf{r_1} + \mathbf{r_2}) \end{bmatrix}$$

$$\langle (\mathbf{c}_1 + \mathbf{c}_2), \mathbf{s} \rangle = 2\mathbf{e}^T(\mathbf{r}_1 + \mathbf{r}_2)$$

Multiplication: $\mathbf{c}_1 \otimes \mathbf{c}_2$ encrypts $m_1 \cdot m_2$ under the *new* key $\mathbf{s} \otimes \mathbf{s}$

$$m_1 \cdot m_2 = [[\langle \mathbf{c}_1 \otimes \mathbf{c}_2, \mathbf{s} \otimes \mathbf{s} \rangle]_q]_2$$

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

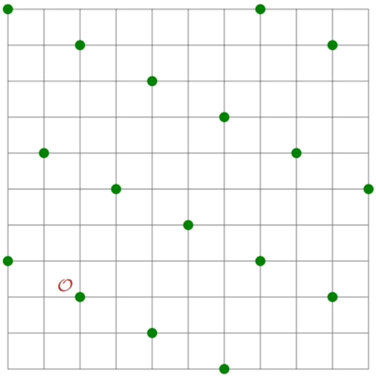
Also, how do we show that LWE problem is hard?

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

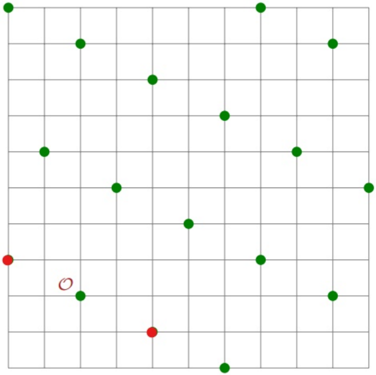
These problems are conjectured to be both classically and quantum hard

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

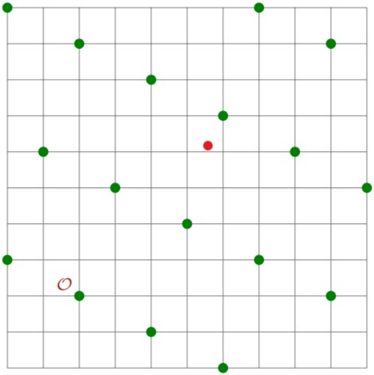
These problems are conjectured to be both classically and quantum hard

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 $\mathrm{Enc}(a)$ and $\mathrm{Enc}(b)$ we want $\mathrm{Enc}(a + b)$ and $\mathrm{Enc}(a \cdot b)$

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 $\text{Enc}(a)$ and $\text{Enc}(b)$ we want $\text{Enc}(a+b)$ and $\text{Enc}(a \cdot b)$

Homomorphic Encryption does not exist with traditional crypto tools

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

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

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 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:

$$m \xrightarrow{\quad\text{RSA}\quad} c$$
$$m \xrightarrow{\quad\text{RSA}\quad} c$$

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:

$$m \xrightarrow{\text{RSA}} c$$
$$m \xrightarrow{\text{RSA}} c$$

$$m \xrightarrow{\text{LC}} c_1 + e_1$$
$$m \xrightarrow{\text{LC}} c_1 + e_2$$

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:

$$m \xrightarrow{\text{RSA}} c$$
$$m \xrightarrow{\text{RSA}} c$$

$$m \xrightarrow{\text{LC}} c_1 + e_1$$
$$m \xrightarrow{\text{LC}} c_1 + e_2$$

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?

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.

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.

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 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

(Ring) LWE Works Cited

1. **Regular LWE:**
   [Reg05] O. Regev. *On lattices, learning with errors, random linear codes, and cryptography*. In STOC, H. N. Gabow and R. Fagin, eds., ACM, New York, 2005, pp. 84–93.
2. **RLWE:**
   [LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. *On ideal lattices and learning with errors over rings*. In EUROCRYPT, Springer, Berlin, 2010, pp. 1–23

Fully Homomorphic Encryption Schemes

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

   [G09] Craig Gentry. *Fully homomorphic encryption using ideal lattices*. In Michael Mitzenmacher,ed., *STOC*, pages 169-178. ACM, 2009.

2. **RLWE Schemes**:

   1. FHE without bootstrapping:

   [BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. *Fully homomorphic encryption without bootstrapping*. In ITCS, S. Goldwasser, ed., ACM, New York, 2012, pp. 309–325

   2. FHE Batching:

   [GHS12] S. Halevi, and N. P. Smart, *Fully homomorphic encryption with polylog overhead*. In EUROCRYPT, Lecture Notes in Comput. Sci. 7237, D. Pointcheval and T. Johansson, eds., Springer, Heidelberg, 2012, pp. 465–482