Introduction to NIST's Module-Lattice-Based Key-Encapsulation Mechanism standard (ML-KEM)

Anna Jokiniemi

NOKIA

Contents

- Threat of quantum computers to cryptography
- ML-KEM generally
- Key-Encapsulation Mechanisms (KEM)
- Module-Lattice problems (ML)
- Provable security

Cryptography

- Science of encrypting and decrypting data, so that only the intended recipient can access it
- For instance, phones handle a large amount of sensitive information and rely on various cryptographic methods
- This field is crucial, and we should all take an interest in it

Threat posed by quantum computers

- Sufficiently powerful quantum computers may become a reality within the next 20 years
- They can break many modern cryptographic methods
- Once such quantum computers are available, a significant amount of personal data could be at risk

20 years sounds like a long time, but…

- Harvest now, decrypt-later attacks
  - The adversary stores data encrypted using today's methods, and decrypts it once quantum computers become available
- If the data encrypted today remains sensitive in the future, it must be encrypted using quantum-resistant methods

20 years sounds like a long time, but…

- Transitioning to quantum-resistant methods takes time as it involves creating new methods, testing and implementing them
- It is critical to begin the preparation and development of quantum-safe solutions now

Cryptographic standards

- Historically, cryptographic methods were considered secure if the designer could not find any vulnerabilities
  - NOT A GOOD STRATEGY! Adversaries can be cleverer than the designers
- Instead, cryptographic methods should be studied by multiple scholars, and a trusted organization should evaluate and standardize them

The National Institute of Standards and Technology(NIST)

- A large agency of the United States Department of Commerce
- Provides cryptographic standards that are very widely used
  - Pretty much all other security organizations follow their recommendations
  - Their standards have a significant impact on cryptographic methods

ML-KEM generally

- NIST takes the quantum threat seriously and has recently standardized new quantum-safe mechanisms
- One of these quantum-safe standards is ML-KEM
- ML-KEM is a quantum-safe cryptographic method standardized by NIST

Symmetric encryption

- Alice wants to send a message to Bob securely
- They use a symmetric key for both encryption and decryption
- Alice generates a key and uses it to encrypt the message
- She then gives both the key and the encrypted message to Bob
- Since Bob has the key, he can use it to decrypt the message

Symmetric encryption

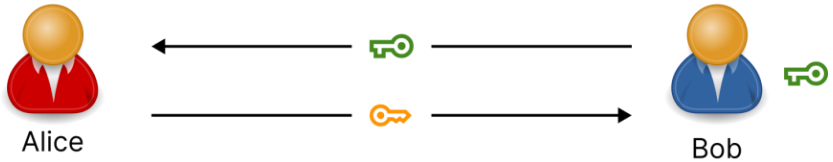
- Computationally efficient encryption method
- However, we need a secure way for Alice to transfer the key to Bob
  - For example, they could meet in person
- In many cases, there is no secure way to do this

Public key encryption

- We use a pair of mathematically linked keys
  - A public key for encryption
  - A secret key for decryption
- Bob generates a pair of keys: he makes one public (maybe on website) and keeps the secret key private
- Alice obtains Bob's public key and uses it to encrypt message
- Only Bob's secret key can decrypt the message

Key encapsulation mechanisms (KEM)

- Unfortunately, public key encryption methods are inefficient for large messages
- Instead, a combination of both methods is used
- Alice generates a symmetric key and uses it to encrypt a message
- She then sends the symmetric key to Bob using the public key encryption method
- Bob can now decrypt the symmetric key and use it to decrypt the message

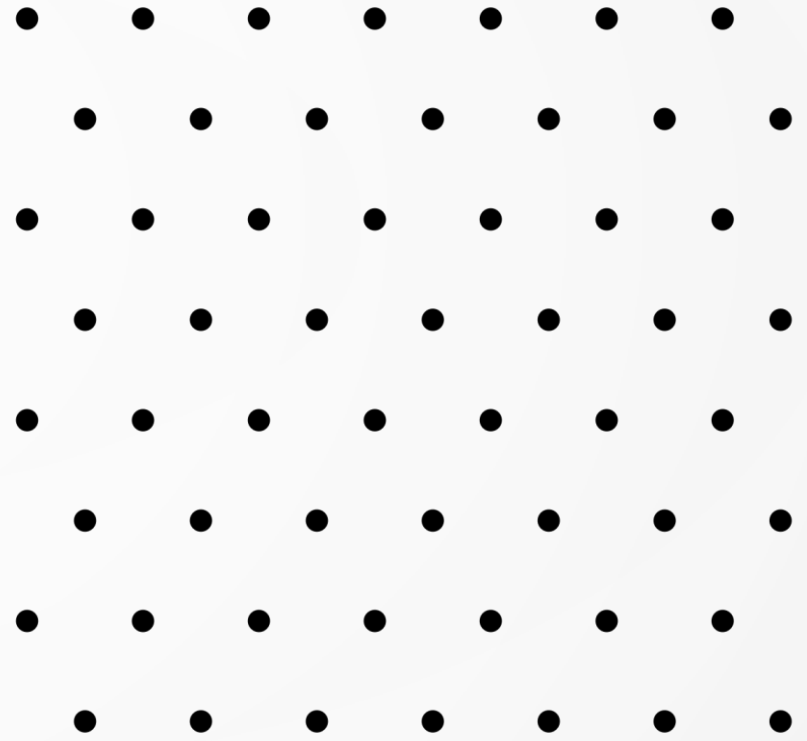


Hard problems

- All cryptographic methods rely on problems that are assumed to be hard to solve without a key
- If you have the key, the problem becomes easy to solve
- ML-KEM is assumed to be secure if a module-lattice problem, known as the module learning with errors problem, is hard to solve for both classical and quantum computers

Module-Lattices (ML)

- A lattice is a set of points with a regular and repeating structure
- A module is a specific algebraic structure
- A module-lattice is a lattice on which a module structure has been imposed
- There are several problems related to module-lattices that are assumed to be hard for both classical and quantum computers



Module Learning With Errors

- A module-lattice problem that is assumed to be hard
- ML-KEM is based on this problem
- We will use polynomials of the form $f = f_0 + f_1X + \cdots + f_{255}X^{255}$, where the coefficients f_i are sampled from the set $\{0,1,2,\cdots,q-1\}$ for some q
- Vectors and matrices whose elements are polynomials of the above form are elements of a module

Parameters of Module Learning With Errors problem

- Public parameters set by NIST: q is a prime, and n and m are integers
- We select a secret vector $\mathbf{s} = (s_0, s_1 \dots s_n)$, where each element is a polynomial of the previously defined form
- We sample a matrix $\boldsymbol{A} = \begin{bmatrix} a_{0,0} & \cdots & a_{0,n} \\ \vdots & \ddots & \vdots \\ a_{m,0} & \cdots & a_{m,n} \end{bmatrix}$, where each element is uniformly sampled polynomial of the previously defined form. The matrix $\boldsymbol{A}$ is made public

Parameters of Module Learning With Errors

- We form an error vector $\boldsymbol{e} = (e_0, \ldots, e_m)$, where elements are 255-degree polynomials, whose coefficients are sampled such that most of them are small numbers from the set $\{0,1,2,\cdots,q-1\}$
- Error vector is kept secret

Module Learning With Errors equation

- We form an equation with an error of the form

$$\boldsymbol{s} \cdot \boldsymbol{A} + \boldsymbol{e} = \boldsymbol{c} \quad (mod\ X^{255} + 1)$$

- The vector $\boldsymbol{c}$ is made public

Module Learning With Errors equation

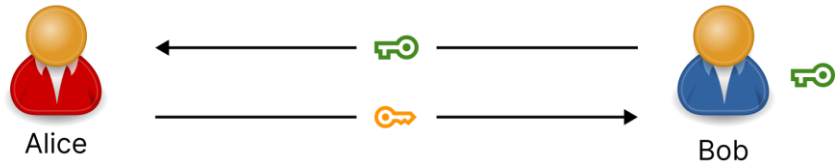
- Everything in the equation is public except for $\boldsymbol{s}$ and $\boldsymbol{e}$

$$\boldsymbol{s} \cdot \boldsymbol{A} + \boldsymbol{e} = \boldsymbol{c} \quad (mod\ X^{255} + 1)$$

- It is assumed to be very hard to solve $\boldsymbol{e}$ without knowing $\mathbf{s}$, even with a quantum computer
- But if $\mathbf{s}$ is known, it becomes easy to solve $\boldsymbol{e}$
- As long this assumption about hardness holds, ML-KEM can be considered secure

ML-KEM's working principle [1]

- Let's form a KEM based on this hard problem
- Alice can hide the symmetric key she wants to send to Bob in the error vector, and only Bob with the secret vector, can retrieve it
- Now, you understand what ML-KEM is and its simplified working principle



Why we only assume the problem to be hard to solve?

- You may notice that we always say a problem is **assumed** to be hard
- It is impossible to prove that something cannot be solved; we can only prove that it can be solved by solving it
- But we must at least aim to model the hardness of the problem, so that we get some sense on its hardness

Provable security

- We model the hardness of a problem
- Modelling requires assumptions and simplifications, as with modeling any real-life phenomena
- Over the past 20 years, this field has faced criticism for the simplifications and presenting findings as if the methods were proven secure
- Today, empirical evidence is preferred

The two most widely used security notions. Added only for illustration [2]

GAME IND-CPA

$(pk, sk) \leftarrow \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$(m_0^*, m_1^*, st) \leftarrow \mathsf{A}_1(pk)$
$c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$
$b' \leftarrow \mathsf{A}_2(pk, c^*, st)$
return $[\![b' = b]\!]$

GAME IND-CCA

$(pk, sk) \leftarrow \mathsf{Gen}$
$b \xleftarrow{\$} \{0,1\}$
$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$
$K_1^* \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathsf{A}^{\textsc{Decaps}}(c^*, K_b^*)$
return $[\![b' = b]\!]$

Provable security of ML-KEM

- Because empirical evidence is favored, the documentation [3] on provable security is short and lacks intermediate steps, making it hard to follow
- My thesis aims to clarify the provable security of ML-KEM

The two most widely used security notions. Added only for illustration [2]

GAME IND-CPA

01 $(pk, sk) \leftarrow \mathsf{Gen}$
02 $b \xleftarrow{\$} \{0,1\}$
03 $(m_0^*, m_1^*, st) \leftarrow \mathsf{A}_1(pk)$
04 $c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$
05 $b' \leftarrow \mathsf{A}_2(pk, c^*, st)$
06 **return** $[\![b' = b]\!]$

GAME IND-CCA

07 $(pk, sk) \leftarrow \mathsf{Gen}$
08 $b \xleftarrow{\$} \{0,1\}$
09 $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$
10 $K_1^* \xleftarrow{\$} \mathcal{K}$
11 $b' \leftarrow \mathsf{A}^{\textsc{Decaps}}(c^*, K_b^*)$
12 **return** $[\![b' = b]\!]$

Summary

- Threat of quantum computers
- Introduced ML-KEM and how it works
- Discussed provable security and my thesis topic

THANK YOU FOR LISTENING!

Sources:

[1] National Institute of Standards and Technology. FIPS 203: Federal Information Processing Standards Publication Module-Lattice-based Key-Encapsulation Mechanism Standard. Available online at https://doi.org/10.6028/NIST.FIPS.203.ipd. Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8900. Aug. 2024.

[2] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A Modular Analysis of the Fujisaki-Okamoto Transformation". In: Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I. Ed. by Yael Tauman Kalai and Leonid Reyzin. Vol. 10677. Lecture Notes in Computer Science. Springer, Cham, 2017, pp. 341–371. isbn: 978-3-319-70499-9. doi: 10.1007/978-3-319-70500-2_12

[3] Roberto Avanzi et al. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.01). Version 3.01. Jan. 2021. url: https : / / pq - crystals.org/kyber/data/kyber-specification-2021-01.pdf