Cryptanalysis on Lattice-Based Cryptography

Students:

- Vũ Tiến Giáp - 22520367
- Nguyễn Viết Duy - 22520336

Lecturer: Nguyễn Ngọc Tự

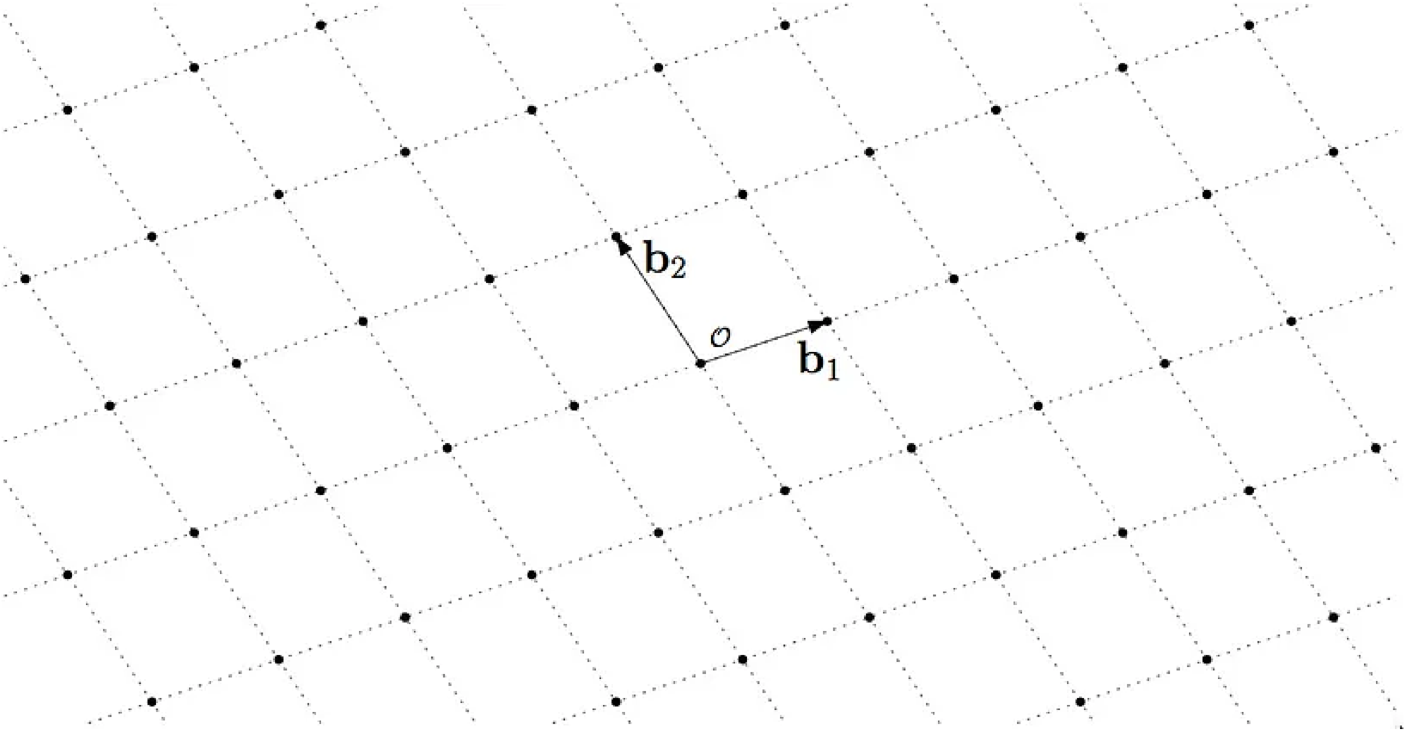
Overview

In the ever-evolving landscape of technology, the government agency recognizes the imperative to fortify its communication infrastructure against the looming threat of quantum advancements. As quantum computers inch closer to reality, the conventional cryptographic protocols currently safeguarding sensitive governmental information become susceptible to rapid decryption. In response to this imminent challenge, the agency contemplates the adoption of lattice-based cryptographic algorithms for post-quantum secure communications.

Lattice-based cryptography offers a unique and promising solution tailored to the agency’s security needs. The inherent resilience of lattice-based algorithms to quantum attacks aligns seamlessly with the agency’s mandate to ensure the confidentiality and integrity of classified information, even in the face of quantum computational capabilities. In this project, we will talk about lattice - introduction, lattice reduction algorithms and lattice problems. We will also discuss about Learning With Error (LWE) problems and cryptosystems using it

Mathematical Background

What is lattice ?



A **lattice** [1] is a set of points in n-dimensional space with a periodic structure, such as the one illustrated in Figure 1. More formally, given n-linearly independent vectors $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$, the lattice generated by them is the set of vectors

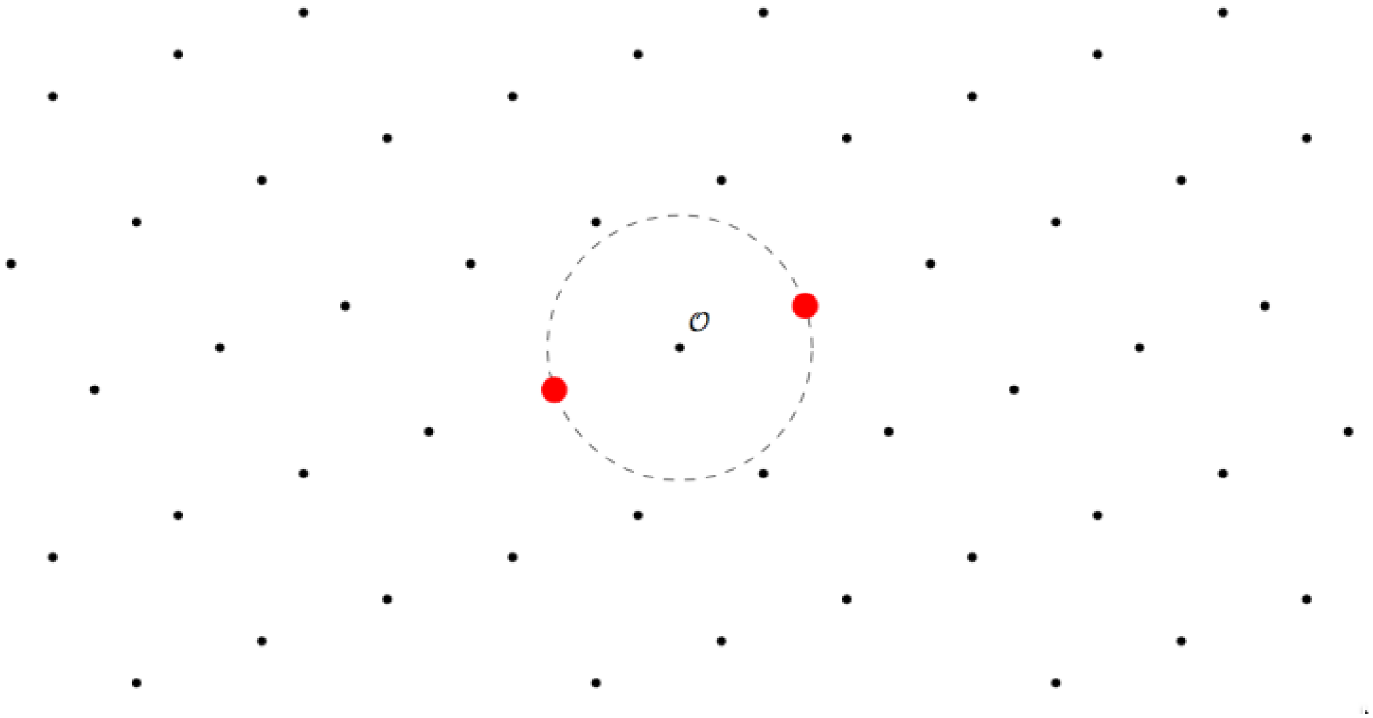
$$\mathcal{L}(b_1, b_2, \ldots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}$$

The vectors $b_1, b_2, \ldots, b_n$ are known as a **basis**
of the lattice. A basis can be represented by the matrix $\mathbf{B = [b_1, b_2, \ldots, b_n]} \in \mathbb{R}^{\mathbf{n \times n}}$ having the basis vectors as columns. Using matrix notation, the lattice generated by a matrix $\mathbf{B} \in \mathbb{R}^{\mathbf{n \times n}}$ can be defined as $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx : x} \in \mathbb{Z}^{\mathbf{n}}\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication

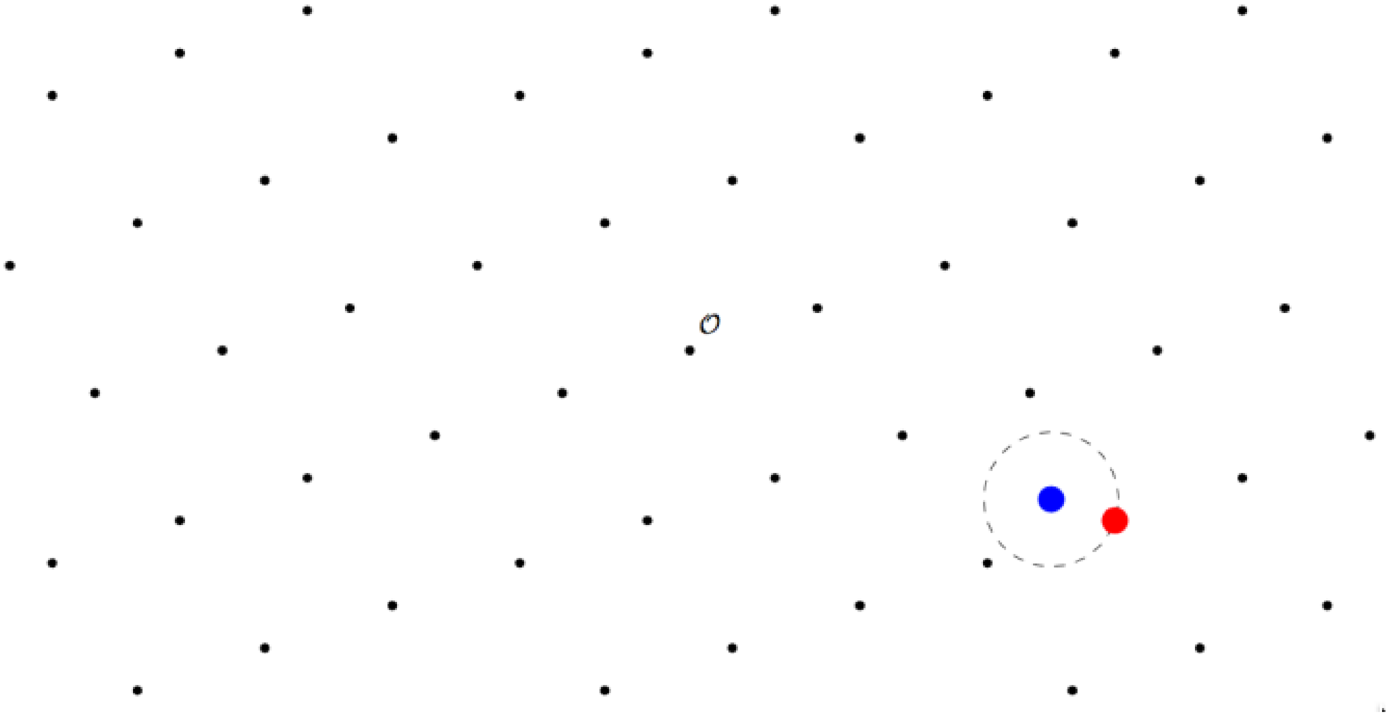
Lattice problems

The most well known computational problems on lattices are the following:

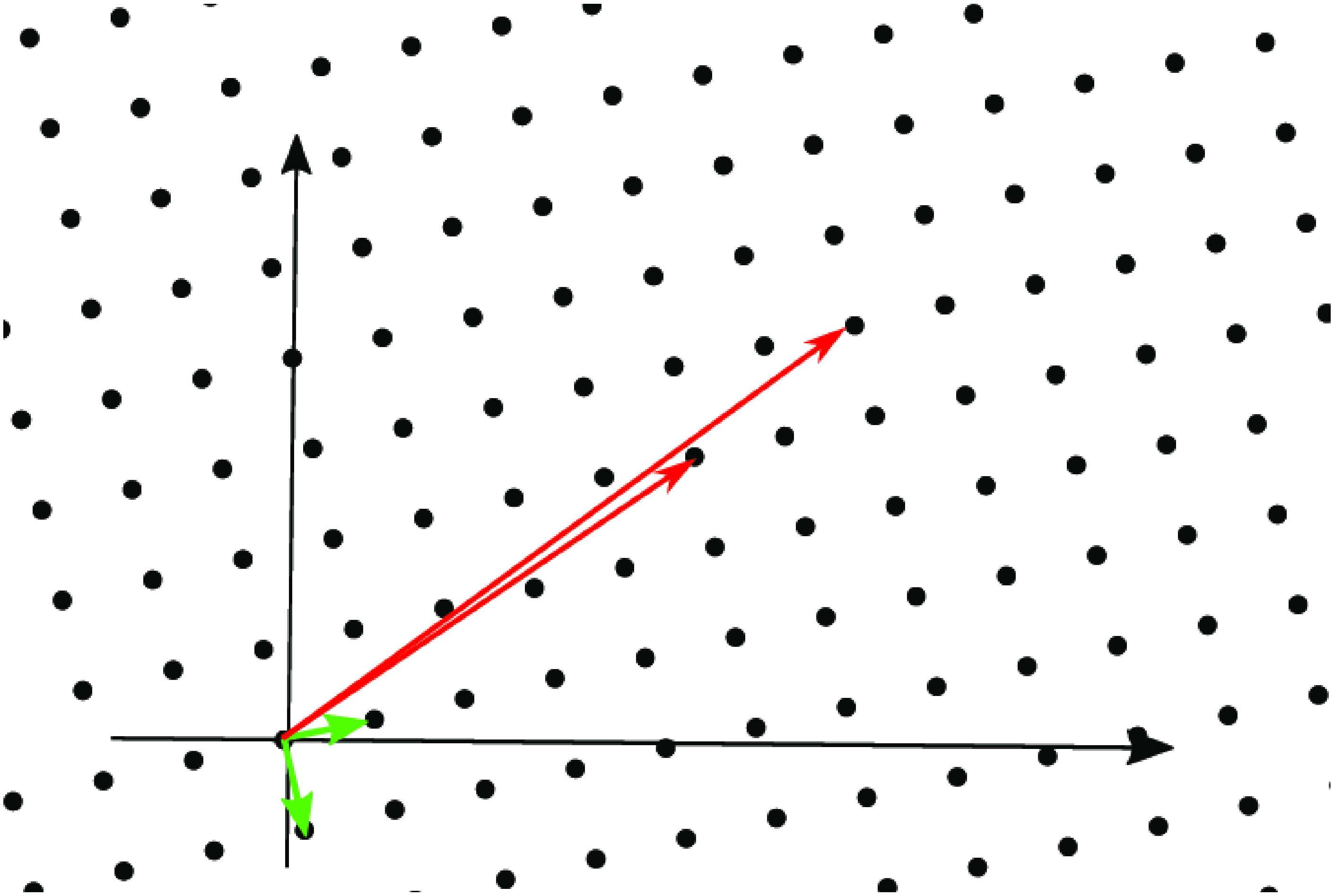
- **Shortest Vector Problem (SVP)**: Given a lattice basis $\mathbf{B}$, find the shortest nonzero vector in $\mathcal{L}(\mathbf{B})$



- **Closest Vector Problem (CVP)**: Given a lattice basis $\mathbf{B}$ and a target vector $\mathbf{t}$ (not necessarily in the lattice), find the lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ closest to $\mathbf{t}$



- **Shortest Independent Vectors Problem (SIVP)**: Given a lattice basis $\mathbf{B} \in \mathbb{R}^{\mathbf{n} \times \mathbf{n}}$, find n linearity independent lattice vectors $\mathbf{S} = [\mathbf{s_1, s_2, \ldots, s_n}]$ (where $s_i \in \mathcal{L}(\mathbf{B})$ for all i) so that $max||v_i|| \leq max||b_i||$, where $||x|| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$



Learning With Error (LWE)

We will give an introduction of LWE:

Definition: Let n, q be positive integers, $\mathcal{X}$ be a probability distribution on $\mathbb{Z}$ and $\mathbf{s}$ be a uniformly random vector in $\mathbb{Z}_q^n$. We denote by $L_{s,\mathcal{X}}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_\mathbf{q}^\mathbf{n}$ uniformly at random, choosing $\mathbf{e} \in \mathbb{Z}$ according to $\mathcal{X}$ and considering it in $\mathbb{Z}_q$, and returning $(\mathbf{a}, \mathbf{c}) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + \mathbf{e}) \in \mathbb{Z}_\mathbf{q}^\mathbf{n} \times \mathbb{Z}_\mathbf{q}$ sampled according to $L_{s,\mathcal{X}}$

We have two computational problems in LWE:

Decision LWE: The problem of deciding whether pairs $(\mathbf{a}, \mathbf{c}) \in \mathbb{Z}_\mathbf{q}^\mathbf{n} \times \mathbb{Z}_\mathbf{q}$ are sampled according to $L_{s,\mathcal{X}}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$

Search LWE: The problem of recovering $\mathbf{s}$ from pairs $(\mathbf{a}, \mathbf{c}) \in \mathbb{Z}_\mathbf{q}^\mathbf{n} \times \mathbb{Z}_\mathbf{q}$ sampled according to $L_{s,\mathcal{X}}$

In this project, we will implicitly assume that $\mathcal{X}$ is centered, i.e. has expectation 0. We may also write LWE in matrix form as $\mathbf{A} * \mathbf{s} + \mathbf{e} = \mathbf{c} \mod q$

The hardness of LWE problems are the same as SVP and CVP

Proposed Solution

Solution Architecture

Key Generation and Collection

We will focus on the LWE-based public key cryptosystem:

Parameter: We let n be the security parameter of the cryptosystem. The cryptosystem is parameterized by two integers m, q and a probability distribution $\mathcal{X}$ on $\mathbb{Z}_q$. The setting of these paramteters that guarantees both security and correctness is the following:

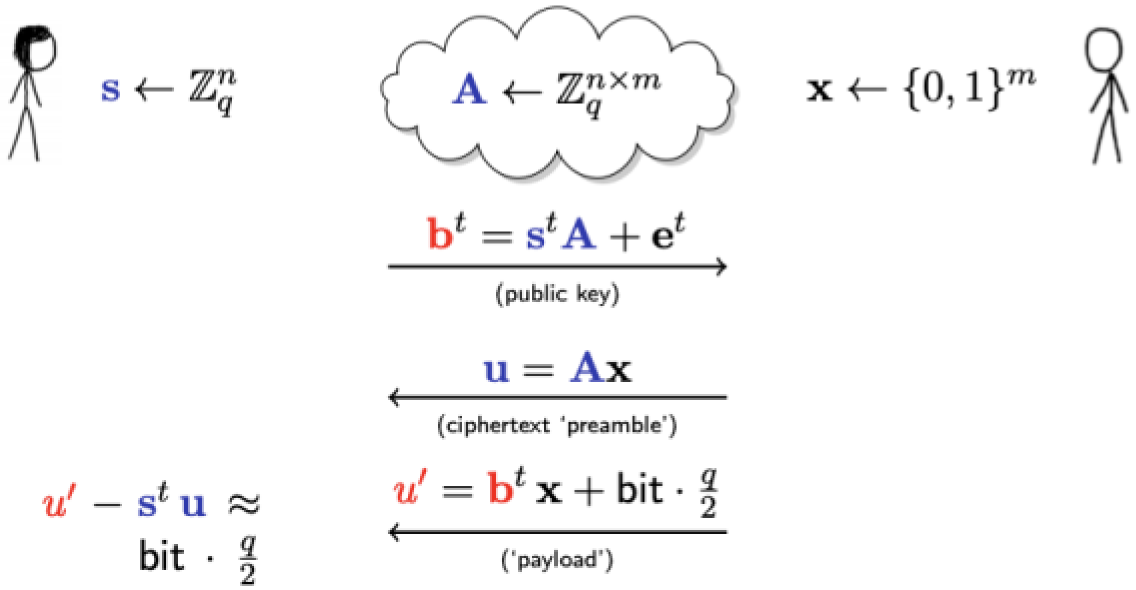
- Prime $q > 2 \in (n^2, 2n^2)$
- $m = (1 + \varepsilon)(n + 1) \log q$ for some arbitrary constant $\varepsilon > 0$
- The probability distribution $\mathcal{X}$ is taken to be $\Psi_{\alpha(n)}$ for $\alpha(n) = o(1/\sqrt{n} \log n)$, i.e., $\alpha(n)$ is such that $\lim\limits_{n \to \infty} \alpha(n) * \sqrt{n} \log n = 0$. For example, we can choose $\alpha(n) = 1/(\sqrt{n} \log n)$

Private Key: Choose $\mathbf{s} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$ uniformly at random. The private key is $\mathbf{s}$

Public Key: For $i = 1, \ldots m$, choose m vectors $a_1, a_2, \ldots, a_m \in \mathbb{Z}_q^n$ independently from the uniform distribution. Also choose elements $e_1, e_2, \ldots, e_m \in \mathbb{Z}_q$ independently according to $\mathcal{X}$. The public key is given by $(\mathbf{a}_i, b_i)_{i=1}^m$ where $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$

Encryption: In order to encrypt a bit we choose a random set S uniformly among all 2^m subsets of $[m]$. The encryption is $(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i)$ if the bit is 0 and $(\sum_{i \in S} \mathbf{a}_i, [\frac{q}{2}] + \sum_{i \in S} b_i)$ if the bit is 1

Decryption: The decryption of a pair $(\mathbf{a}, b)$ is 0 if $b - \langle \mathbf{a}, s \rangle$ is closer to 0 than to $[\frac{q}{2}]$ modulo q. Otherwise, the decryption is 1


$\mathbf{s} \leftarrow \mathbb{Z}_q^n$
$\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$
$\mathbf{x} \leftarrow \{0,1\}^m$
$\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$
(public key)
$\mathbf{u} = \mathbf{A}\mathbf{x}$
(ciphertext 'preamble')
$u' = \mathbf{b}^t\mathbf{x} + \text{bit} \cdot \frac{q}{2}$
('payload')
$u' - \mathbf{s}^t\mathbf{u} \approx \text{bit} \cdot \frac{q}{2}$

Cryptanalysis Tools

- C/C++, Python 3.x

- Number Theory Library (NTL): NTL is a C++ library for doing number theory. NTL supports arbitrary length integer and arbitrary precision floating point arithmetic, finite fields, vectors, matrices, polynomials, lattice basis reduction and basic linear algebra. NTL is free software released under the GNU Lesser General Public License v2.1.

- Sagemath: SageMath is a computer algebra system with features covering many aspects of mathematics, including algebra, combinatorics, graph theory, group theory, differentiable manifolds, numerical analysis, number theory, calculus and statistics. Stein realized when designing Sage that there were many open-source mathematics software packages already written in different languages, namely C, C++, Common Lisp, Fortran and Python.

- Numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Attack Models

We will focus on LWE cryptosystem [2]. When the parameters didn't verify condition at **Key Generation and Collection** part, this cryptosystem is vulnerable.

I. Algebraic Algorithm:

Arora-Ge

This is an attack due to Arora and Ge [3]. The basic idea is to view a LWE sample $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ where $e \in S \subseteq \mathbb{Z}_q$ as a polynomial equation

$$f_{\mathbf{a},b}(\mathbf{s}) = \prod_{x \in S} (b - \langle \mathbf{a}, \mathbf{s*} \rangle - x) \mod q$$

where $b, \mathbf{a}$ are known and $\mathbf{s}$ is treated as the unknown variable (denoted by s^*). Clearly, if $(\mathbf{a}, b)$ is an LWE sample, then $f_{\mathbf{a},b}(\mathbf{s}) = 0 \mod q$, else it isn't. Solving the system of polynomial equations

$$\{f_{\mathbf{a}_i,b_i}(\mathbf{s}) = 0 \mod q\}_{i=1}^{m}$$

of degree $|S|$ will give us the LWE secret.

When $\mathcal{X}$ is the discrete Gaussian distribution. Let's now see what this does to $LWE(n, m, q, \mathcal{X})$ where $\mathcal{X}$ is a Gaussian with standard deviation s. The probability that the error parameter is less than $k.s$ is $e^{-O(k^2)}$

- We get a reasonable chance that all equations have error bounded by $k.s$ if $m.e^{-O(k^2)} \ll 1$.

- On the other hand, we need $m > \binom{n}{k.s}$ for linearization to work.

II. Combinatorial Algorithm

Blum-Kalai-Wasserman

This is an attack originally due to Blum, Kalai and Wasserman[4]. A similar version was later discovered by Wagner[5].

The basic idea is to find small-weight linear combinations $\mathbf{x}_{i,j}$ of the columns of $\mathbf{A}$ that sums up to the fixed vector, say the unit vectors $\mathbf{u}_i$, that is $\mathbf{A}\mathbf{x}_{i,j} = \mathbf{u}_i \mod q$. Once we find such vectors, we compute

$$\mathbf{b}^T\mathbf{x}_{i,j} = (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T)\mathbf{x}_{i,j} = s_i + \mathbf{e}^T\mathbf{x}_{i,j} \mod q$$

which, give many copies and averaging, gives us s_i as long as $||\mathbf{e}^T\mathbf{x}_{i,j}|| \ll q$. Iterating for all $i = 1, .., n$ gives us s.

The BKW algorithm – when applied to Search-LWE – can be viewed as consisting of three stages somewhat analogous to those of linear system solving:

1. Sample reduction is a form of Gaussian elimination which, instead of treating each component independently, considers 'blocks' of b components per iteration, where b is a parameter of the algorithm.
2. Hypothesis testing tests candidate sub-solutions to recover components of the secret vector s.
3. Back substitution such that the whole process can be continued on a smaller LWE instance.

III. A Geometric (Suite of) Algorithm(s)

Lattice Reduction

This is an attack that follows using the LLL algorithm[6] and (building on LLL) the BKZ algorithm[7] that find approximately short vectors in integer lattices.

The attack use the fact that LWE is, at its core, a problem of finding short vectors in integer lattices. Taking n LWE samples $(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$, and write them in matrix form as $\mathbf{As} + \mathbf{e} = \mathbf{b} \mod q$, where $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2}, \mathbf{...}, \mathbf{a_n}\}$, $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$

Now, the vector $\mathbf{As}$ could be interpreted as a lattice point in the lattice $\mathcal{L}(\mathbf{a_1}, \mathbf{a_2}, \mathbf{...}, \mathbf{a_n})$ with unknown coefficient defined by the secret vector $\mathbf{s}$, and $\mathbf{e}$ is small so $\mathbf{b}$ is pretty close to this lattice point. This corresponds to a CVP problem, where you want to find the closest vector to $\mathbf{b}$ in this lattice.

Primal Attack

The primal attack[8] is a kind of classical and useful attack model for the search-LWE problem and it only requires polynomial LWE samples. The core idea is that transforming the search-LWE instance into a unique-SVP and solving the unique shortest vector by lattice reduction with appropriate root-Hermite factor δ. We recap this model as follows.

Given an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$ with matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$, there are three types of embedding techniques to reduce search-LWE problem: Kannan's embedding, dual embedding and Bai-Galbraith embedding

(1). The Kannan's embedding[9] reduces the BDD problem to SVP. The corresponding embedding lattice is

$$\mathcal{L}_K = \left\{ \mathbf{y} \in \mathbb{Z}^{m+1} : \mathbf{y} = \mathbf{A^*x} \mod q, \forall \mathbf{x} \in \mathbb{Z}^{n+1}, \mathbf{A^*} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mu \end{bmatrix} \in \mathbb{Z}^{(m+1)\times(n+1)} \right\}$$

and in practice it is preferable to use $\mu = 1$. A short vector in lattice is $\mathbf{v} = (\mathbf{e} \mid \mathbf{1})$.

(2). The dual embedding proposed by Bai and Galbraith [10] constructs a lattice related to both secret $\mathbf{s}$ and error $\mathbf{e}$. The corresponding embedding lattice is:

$$\mathcal{L}_D = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} : (\mathbf{I}_m|\mathbf{A}| - \mathbf{b})\mathbf{x} = \mathbf{0} \mod q\},$$

which has a basis

$$\mathbf{B} = \begin{bmatrix} q\mathbf{I}_m & -\mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}$$

The vector $\mathbf{v} = (\mathbf{e} \mid \mathbf{s} \mid \mathbf{1})$ is a short vector in lattice

(3). The Bai-Galbraith embedding improves dual embedding for such LWE instance that secret and error are chosen from different distributions, its core idea is to balance the size of the error and the secret. Specially, the short vector in lattice $\mathcal{L}_D$ can be re-balanced as $\mathbf{v} = (\mathbf{e} \mid \omega\mathbf{s} \mid \omega)$ with scaling factor $\omega = \frac{\sigma_e}{\sigma_s}$ and the new embedding lattice is:

$$\mathcal{L}_\omega = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} : (\mathbf{I}_m|\frac{1}{\omega}\mathbf{A}| - \frac{1}{\omega}\mathbf{b})\mathbf{x} = \mathbf{0} \mod q\},$$

Dual attack

The dual attack[11][12], which is a distinguishing attack, can be use for solving **Decision LWE** problem by this strategy:
Given samples $(\mathbf{A}, \mathbf{b})$:

1. Find many short vectors $\mathbf{x}$ such that $\mathbf{y} = \mathbf{x}\mathbf{A}$ are also short
2. Compute $\langle \mathbf{x}, \mathbf{b} \rangle$.

- Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ or $\mathbf{b}$ uniformly random:
  - If $\mathbf{b}$ is uniformly random, so is $\langle \mathbf{x}, \mathbf{b} \rangle$.
  - If $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, then $\langle \mathbf{x}, \mathbf{b} \rangle = \langle \mathbf{x} \cdot \mathbf{A}, \mathbf{s} \rangle + \langle \mathbf{x}, \mathbf{e} \rangle \equiv \langle \mathbf{x}, \mathbf{e} \rangle \mod q$. If $\mathbf{x}$ is sufficiently short, then $\langle \mathbf{x}, \mathbf{e} \rangle$ will also be short, since $\mathbf{e}$ is also small.

The distinguishing attack can also be used to recover the secret. Given an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$ with matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$, we suppose that $\mathcal{X}_e$ and $\mathcal{X}_s$ are supported on small values with high probability. We find many short vectors $\mathbf{x} \in \mathbb{Z}_q^m$ such that $\mathbf{y} = \mathbf{x}\mathbf{A} \in \mathbb{Z}_q^n$ are also short, and calculate the list of value $(\mathbf{xb})$. For an LWE pair, we have $\mathbf{xb} = \mathbf{ys} + \mathbf{xe}$ which are approximately distributed according to a modular Gaussian distribution. Given sufficiently many samples, we can distinguish between the two distributions.

In order to perform a key recovery attack, we partition $\mathbf{s}$ into two components: $\mathbf{s} = (\mathbf{s}_1 \mid \mathbf{s}_2)$. We partition the matrix $\mathbf{A}$ analogously: $\mathbf{A} = (\mathbf{A}_1 \mid \mathbf{A}_2)$, so that

$$\mathbf{b} = \mathbf{As} + \mathbf{e} = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 + \mathbf{e}$$

If $\mathbf{s}_1$ were known, we could create a new LWE problem

$$\mathbf{b}' = \mathbf{A}_2\mathbf{s}_2 + \mathbf{e}$$

Where $\mathbf{b}' = \mathbf{b} - \mathbf{A}_1\mathbf{s}_1$. However, $\mathbf{s}_1$ is unknown. Nonetheless, we may enumerate over $\mathbf{s}_1$ and use the distinguishing attack on the pairs $(\mathbf{A}_2, \mathbf{b} - \mathbf{A}_1\mathbf{s}_1')$ for every guess $\mathbf{s}_1'$ of $\mathbf{s}_1$ to determine the correct one for which the pairs come from an LWE distribution. In the standard attack, $\mathbf{s}_1$ consists of a single coordinate of s

Summary

Algorithm	**(Some) Broken Parameter setting**
Arora-Ge	$m = \Omega(n^B)$ samples + time where $\lvert Supp(\mathcal{X})\rvert \leq B < q$ [2]
Blum-Kalai-Wasserman	$m < q^{n/\log(q/B)}$ [2]
Lattice Reduction	$m = poly(n, \log q)$ and $q/B = \Omega(2^n)$ and $poly(n, \log q)$ time [2]
Primal Attack	$\sigma\sqrt{b} \leq \sigma_0^{2b-d}.Vol(B)^{\frac{1}{d}}$ [19]
Dual attack	$\mathcal{X}_e$ and $\mathcal{X}_s$ are small

Implementation and Testing

Tool and resources	**Description**
lwe-estimator (https://github.com/malb/lattice-estimator/)	Estimating the concrete security of Learning with Errors instances
Python 3.x	Use to solve some CTF challenges related to LWE, using some techniques below
BKW-Algorithm (https://github.com/AaronHall4/BKW-Algorithm)	An implementation of the Blum-Kalai-Wasserman algorithm for solving the Learning with Errors problem.

Deployment

Lattice-based cryptography is regarded as the rival to a quantum computer attack and the future of post-quantum cryptography. So, cryptographic protocols based on lattices have a variety of benefits, such as security, efficiency, lower energy consumption, and speed

We will see that if the parameters are chosen carefully, there won't be any strategy can attack on LWE, so it is widely used in cryptography to create secure encryption algorithms.

Some selected schemes for the purpose of key exchange, based on LWE problem:

- CRYSTALS-Kyber, [12] which is built upon module learning with errors (module-LWE). Kyber was selected for standardization by the NIST in 2023. [13] In August 2023, NIST published FIPS 203 (Initial Public Draft), and started calling their Kyber version as Module-Lattice-based Key Encapsulation Mechanism (ML-KEM)
- FrodoKEM, [14] a scheme based on the learning with errors (LWE) problem. FrodoKEM joined the standardization call conducted by the National Institute of Standards and Technology (NIST), [11] and lived up to the 3rd round of the process. It was then discarded due to low performance reasons.
- NewHope [15] is based on the ring learning with errors (RLWE) problem.

Reference

[1]. Wikipedia

[2]. https://people.csail.mit.edu/vinodv/CS294/lecture2.pdf (https://people.csail.mit.edu/vinodv/CS294/lecture2.pdf)

[3]. S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, ICALP, volume 6755 of Lecture Notes in Computer Science, pages
403–415. Springer Verlag, 2011.

[4]. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM, 50(4):506–519, 2003.

[5]. Wagner, D. (2002). A Generalized Birthday Problem. In: Yung, M. (eds) Advances in Cryptology — CRYPTO 2002. CRYPTO 2002. Lecture Notes in Computer Science, vol 2442. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45708-9_19 (https://doi.org/10.1007/3-540-45708-9_19)

[6]. Lenstra, A. K.; Lenstra, H. W. Jr.; Lovász, L. (1982). "Factoring polynomials with rational coefficients". Mathematische Annalen. 261 (4): 515–534. CiteSeerX 10.1.1.310.318 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.310.318).

[7]. C.P. Schnorr, A hierarchy of polynomial time lattice basis reduction algorithms, Theoretical Computer Science,Volume 53, Issues 2–3,1987, Pages 201-224,ISSN 0304-3975, https://doi.org/10.1016/0304-3975(87)90064-8 (https://doi.org/10.1016/0304-3975(87)90064-8).

[8]. Xue Zhang; Zhongxiang Zheng; Xiaoyun Wang; (2021). A detailed analysis of primal attack and its variants . Science China Information Sciences, (), –. doi:10.1007/s11432-020-2958-9

[9]. Kannan R. Minkowski's convex body theorem and integer programming. Math Oper Res, 1987, 12: 415–440

[10]. Bai S, Galbraith S D. Lattice decoding attacks on binary LWE. In: Proceedings of Australasian Conference on Information Security and Privacy, 2014. 322–337

[11]. MATZOV. (2022). Report on the Security of LWE: Improved Dual Lattice Attack. Zenodo. https://doi.org/10.5281/zenodo.6493704 (https://doi.org/10.5281/zenodo.6493704)

[12]. Albrecht, M.R. On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL. In Advances inCryptology—EUROCRYPT 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 103–129

[13]. https://pq-crystals.org/ (https://pq-crystals.org/)

[14]. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022 (https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022)

[15]. https://newhopecrypto.org/index.shtml (https://newhopecrypto.org/index.shtml)

[16]. https://csrc.nist.gov/events/2019/second-pqc-standardization-conference (https://csrc.nist.gov/events/2019/second-pqc-standardization-conference)

[17]. https://frodokem.org/ (https://frodokem.org/)

[18]. Martin R. Albrecht, Rachel Player and Sam Scott. On the concrete hardness of Learning with Errors. Journal of Mathematical Cryptology. Volume 9, Issue 3, Pages 169–203, ISSN (Online) 1862-2984, ISSN (Print) 1862-2976 DOI: 10.1515/jmc-2015-0016, October 2015

[19]. https://www.maths.ox.ac.uk/system/files/attachments/lattice-reduction-and-attacks.pdf (https://www.maths.ox.ac.uk/system/files/attachments/lattice-reduction-and-attacks.pdf)