CMSC/Math 456: Cryptography (Fall 2022)

Lecture 25

Daniel Gottesman

Administrative

Problem set #8 is due Dec. 1 at *midnight*.

Problem set #7 solutions are available.

This class is being recorded

Quantum Timelines

Today's quantum computers have ~100 qubits (= quantum bits).

Cryptographic algorithms probably require on the order of 1 million high-fidelity qubits.

How long will this take?

The IBM Quantum roadmap says that they will have 4,000 qubit devices in 2025.

It's certainly possible that quantum computers will be a threat to cryptographic systems in 20 years.

So we need protection against quantum computation for any protocol likely to last that long and for anything that needs to stay secret for that long.

Quantum Algorithms for Crypto

Shor's algorithm: Solves factoring and discrete log efficiently. (The time is limited by the time to perform modular exponentiation.)

> Consequence: RSA and Diffie-Hellman (including with elliptic curves) are insecure against a quantum computer.

Grover's algorithm: Speeds up exhaustive search from $O(N)$ to $O(\sqrt{N})$.

> Consequence: AES and other symmetric cryptosystems need to double key lengths to retain the same level of security against a quantum computer.

Collision-finding: Instead of a birthday attack needing $O(\sqrt{N})$ hash function evaluations, needs $O(N^{1/3})$.

> Consequence: Hash functions need to increase output lengths by x1.5 to retain the same level of security against a quantum computer.

Post-Quantum Cryptography

RSA and Diffie-Hellman are insecure against quantum computers, so we need new public-key encryption protocols or KEMs and new digital signature schemes.

We need protocols whose security is based on different algorithms which can't be broken by known quantum algorithms.

Note: The security of these new cryptographic protocols has not been studied as much as RSA/Diffie-Hellman, so it is more likely that there is an undiscovered classical or quantum attack against them. But at least we don't know how to break them on a quantum computer.

Another reason to study these protocols now is to give more time to try to break them before they are in widespread use.

Current Status

NIST is running a competition for post-quantum cryptography. Currently:

- One KEM protocol and three digital signature protocols selected to become standards.
- Four public key encryption/KEM protocols selected for Round 4 for further study.
  - However, one of these four was broken after being selected for Round 4.
- A new call for additional signature protocols has been issued with a goal of diversifying the types of protocols to standardize.

The goal is not to choose a single standard but rather a portfolio of possible protocols considered secure.

Security and efficiency are both important for this contest.

Encryption/KEM Options

Selected for standardization: Crystals-Kyber, which is based on lattice problems, specifically the learning-with-errors (LWE) problem.

Still under consideration: BIKE, Classic McEliece, and HQC, which are all based on error-correcting codes. The math is fairly similar to lattice problems.

> The idea of error-correcting code protocols is older (Classic McEliece is an evolution of a 1978 proposal), and their security is therefore better understood. However, lattice-based schemes are more efficient and easier to use to give protocols with additional properties (such as homomorphic encryption).

Matrices and Vectors

We will need just the most basic elements of linear algebra to discuss lattice-based encryption protocols.

A matrix M is an $m \times n$ array of numerical (or variable) entries, with M_{ij} the entry in the ith row and jth column.

A vector $\mathbf{v}$ is a list of n numerical (or variable) entries, with v_i the ith entry. There is a standard graphical representation of vectors with n entries as coordinates in n-dimensional space.

Matrix times a vector is a vector: $(M\mathbf{v})_i = \sum_j M_{ij} v_j$.

Vector times a matrix is a vector: $(\mathbf{v}^T M)_j = \sum_i v_i M_{ij}$.

Dot product between 2 vectors is a number: $\mathbf{v} \cdot \mathbf{w} = \mathbf{v}^T \mathbf{w} = \sum_i v_i w_i$.

$$\mathbf{v}^T M \mathbf{w} = \mathbf{v} \cdot (M\mathbf{w}) = \sum_{ij} v_i M_{ij} w_j$$

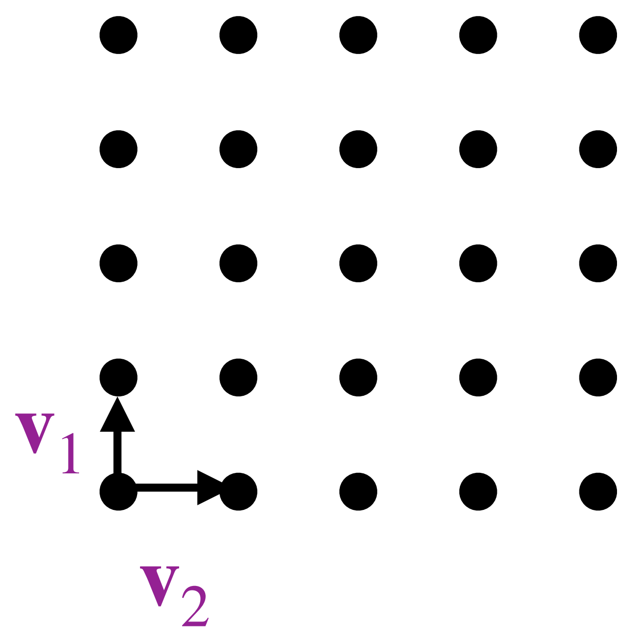
This class is being recorded

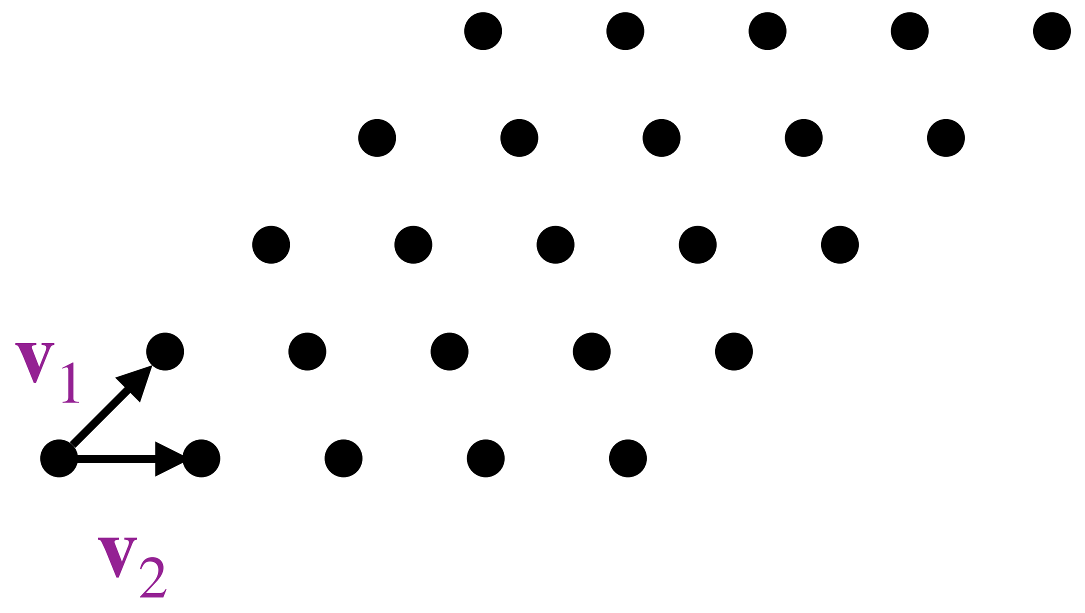
Lattices

A lattice is a regular array of points, integer combinations of some set of vectors.

$$L = \left\{ \sum_i s_i \mathbf{v}_i \mid s_i \in \mathbb{Z} \right\}$$

Examples:

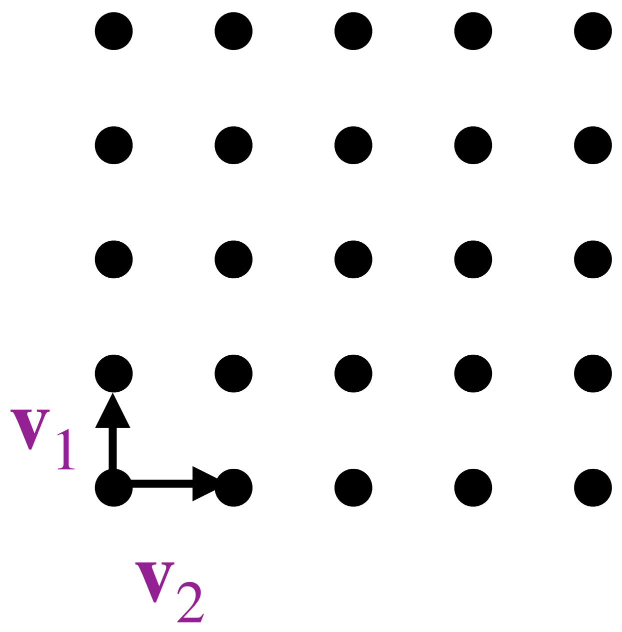


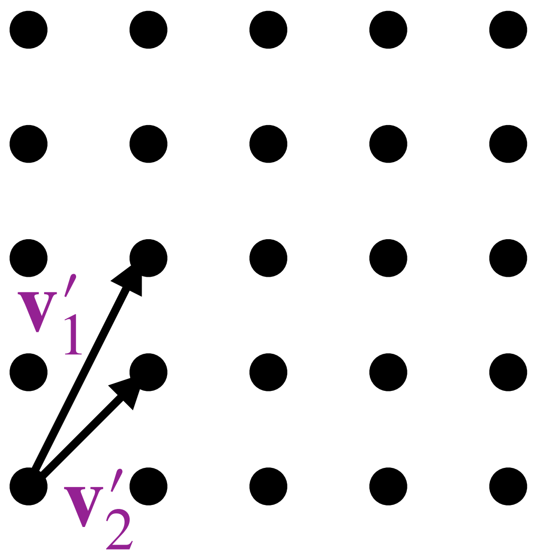


This class is being recorded

Alternate Generating Sets

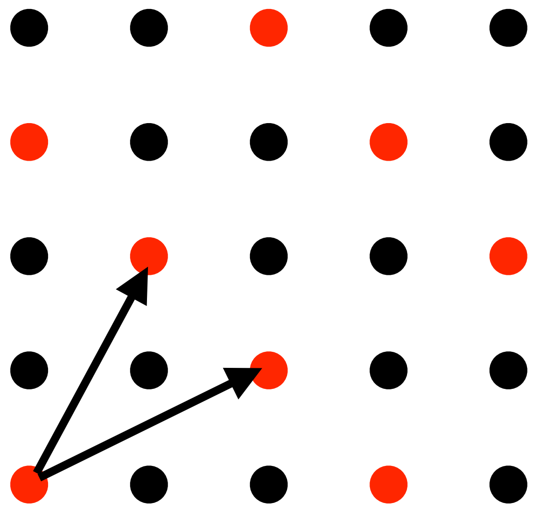
The same lattice can be generated by different sets of vectors.





In this example, $\mathbf{v}_1' = 2\mathbf{v}_1 + \mathbf{v}_2$ and $\mathbf{v}_2' = \mathbf{v}_1 + \mathbf{v}_2$. Also, $\mathbf{v}_1 = \mathbf{v}_1' - \mathbf{v}_2'$ and $\mathbf{v}_2 = 2\mathbf{v}_2' - \mathbf{v}_1'$.

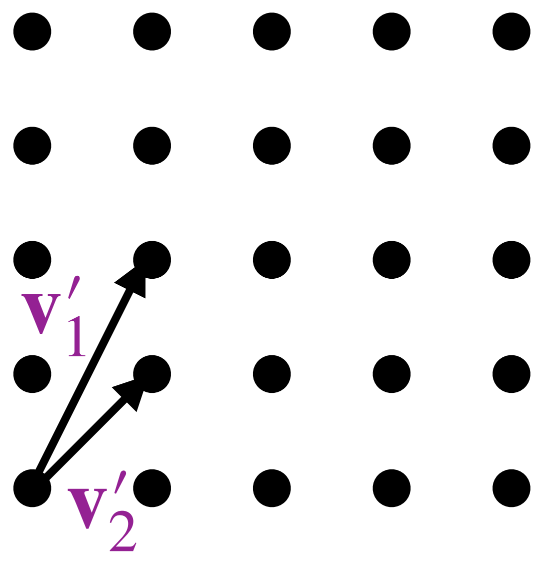
Note that some sets of vectors might generate a sublattice instead.



This class is being recorded

Shortest Vector in a Lattice

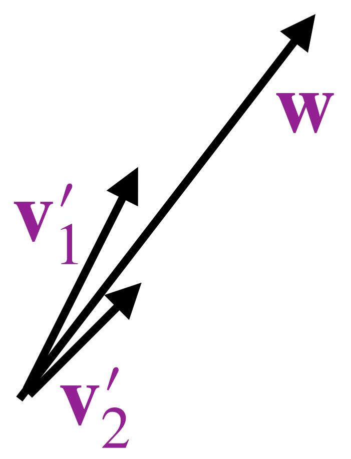
In this example, notice that the shortest vectors in the lattice are $\mathbf{v}_1$ and $\mathbf{v}_2$, which are not part of the generating set.



With a high-dimensional lattice, looking at different integer linear combinations of generating vectors to find the shortest vector(s) is a challenging problem.

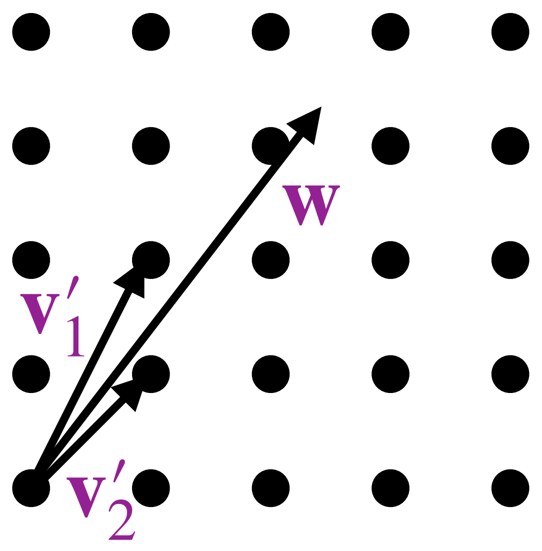
Depending on the properties of the lattice and whether we want to find the absolutely smallest vector or merely one that is an approximation to the smallest vector, this problem can be computationally hard (including against quantum algorithms), up to and including NP-hard.

Closest Lattice Vector



Given $\mathbf{w}$ which is not in the lattice, which lattice vector is it closest to?

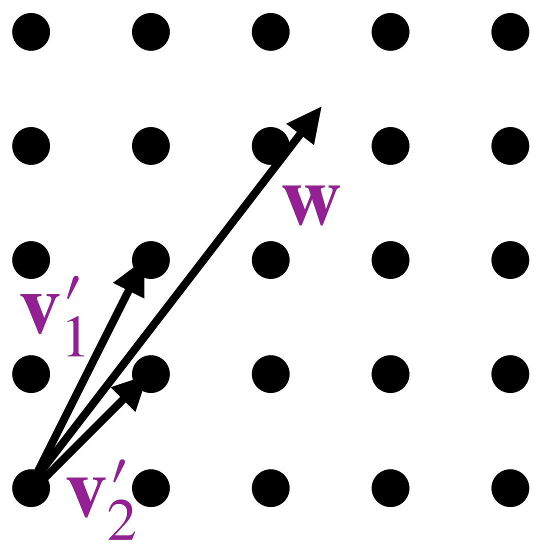
Closest Lattice Vector



Given $\mathbf{w}$ which is not in the lattice, which lattice vector is it closest to?

In this example, $\mathbf{w}$ is closest to $\mathbf{v}'_1 + \mathbf{v}'_2$.

Closest Lattice Vector



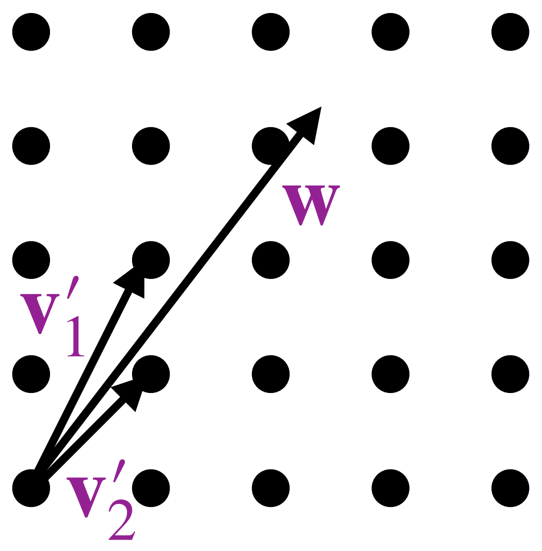
Given $\mathbf{w}$ which is not in the lattice, which lattice vector is it closest to?

In this example, $\mathbf{w}$ is closest to $\mathbf{v}_1' + \mathbf{v}_2'$.

Again, with a high-dimensional lattice, this problem can be computationally hard, depending on properties of the lattice.

We can think of $\mathbf{w}$ as being of the form $\mathbf{v} + \mathbf{e}$, where $\mathbf{v}$ is a lattice vector and $\mathbf{e}$ is a short vector not in the lattice.

Closest Lattice Vector



Given $\mathbf{w}$ which is not in the lattice, which lattice vector is it closest to?

In this example, $\mathbf{w}$ is closest to $\mathbf{v}_1' + \mathbf{v}_2'$.

Again, with a high-dimensional lattice, this problem can be computationally hard, depending on properties of the lattice.

We can think of $\mathbf{w}$ as being of the form $\mathbf{v} + \mathbf{e}$, where $\mathbf{v}$ is a lattice vector and $\mathbf{e}$ is a short vector not in the lattice.

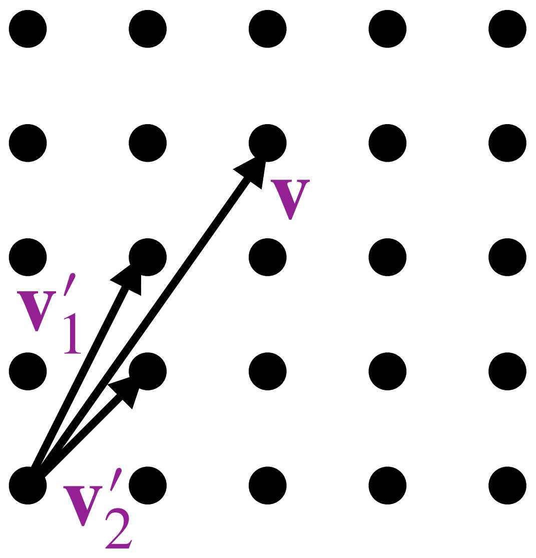
Note: If $\mathbf{w}$ *is* in the lattice, finding which vector it is can be done easily using linear algebra.

Lattice Vectors and Matrices

We can write an arbitrary element of the lattice in the form

$$\mathbf{v} = A\mathbf{s}$$

where A is a matrix where the ith column is the generating vector $\mathbf{v}_i$ and s is the vector of coefficients.



Example: Here,

$$A = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \qquad \mathbf{s} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

so

$$\mathbf{v} = A\mathbf{s} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

This class is being recorded

Learning With Errors

The Learning With Errors (LWE) problem is to solve the closest lattice vector problem with some distribution over $\mathbf{s}$ and $\mathbf{e}$, but with arithmetic mod q.

Specifically: Given matrix A, vector $\mathbf{w}$, find $\mathbf{s}$ such that

$$A\mathbf{s} + \mathbf{e} = \mathbf{w} \bmod q$$

for some "small" $\mathbf{e}$.

Decisional LWE: Determine if $\mathbf{w}$ is uniformly random or if it was generated as

$$A\mathbf{s} + \mathbf{e} = \mathbf{w} \bmod q$$

for some "small" $\mathbf{e}$.

LWE and decisional LWE are of equivalent difficulty.

LWE Encryption

Fixed protocol parameters: m, n, and q, as well as a distribution ψ of short vectors.

Gen: Bob chooses a random $m \times n$ matrix A, an n-dimensional vector $\mathbf{s}$ according to ψ, and an m-dimensional vector $\mathbf{e}$ according to ψ. Let $\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$. The public key is $(A, \mathbf{w})$ and the private key is $\mathbf{s}$.

Enc: Given public key $(A, \mathbf{w})$, to encrypt a single bit b, Alice chooses an m-dimensional vector $\mathbf{r}$ according to ψ, an n-dimensional vector $\mathbf{f}$ according to ψ, and a number x according to ψ. The ciphertext is $(\mathbf{c}, c')$, with

$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$
$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$

LWE Decryption

Dec: Given private key $\mathbf{s}$ and ciphertext $(\mathbf{c}, c')$, Bob computes

$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

Then Bob determines if k is close to 0, in which case Dec outputs b=0, or if k is close to $\lfloor q/2 \rfloor$, in which case Dec outputs b=1.

Here we consider k to be close to 0 if it is slightly larger than 0 or slightly less than q (which can also be interpreted as being a negative number of small absolute value).

LWE Correctness

Recall:

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$$
$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$
$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$
$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

LWE Correctness

Recall:

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$$

$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$

$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$

$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

Then

$$c' = \mathbf{r}^T(A\mathbf{s} + \mathbf{e}) + x + b\lfloor q/2 \rfloor = \mathbf{r}^T A\mathbf{s} + \mathbf{r} \cdot \mathbf{e} + x + b\lfloor q/2 \rfloor \bmod q$$

LWE Correctness

Recall:

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$$
$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$
$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$
$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

Then

$$c' = \mathbf{r}^T(A\mathbf{s} + \mathbf{e}) + x + b\lfloor q/2 \rfloor = \mathbf{r}^T A\mathbf{s} + \mathbf{r} \cdot \mathbf{e} + x + b\lfloor q/2 \rfloor \bmod q$$

and

$$\mathbf{c} \cdot \mathbf{s} = (\mathbf{r}^T A + \mathbf{f}^T)\mathbf{s} = \mathbf{r}^T A\mathbf{s} + \mathbf{f} \cdot \mathbf{s} \bmod q$$

LWE Correctness

Recall:

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$$
$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$
$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$
$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

Then

$$c' = \mathbf{r}^T(A\mathbf{s} + \mathbf{e}) + x + b\lfloor q/2 \rfloor = \mathbf{r}^T A\mathbf{s} + \mathbf{r} \cdot \mathbf{e} + x + b\lfloor q/2 \rfloor \bmod q$$

and

$$\mathbf{c} \cdot \mathbf{s} = (\mathbf{r}^T A + \mathbf{f}^T)\mathbf{s} = \mathbf{r}^T A\mathbf{s} + \mathbf{f} \cdot \mathbf{s} \bmod q$$

so

$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$
$$= b\lfloor q/2 \rfloor + (x + \mathbf{r} \cdot \mathbf{e} - \mathbf{f} \cdot \mathbf{s}) \bmod q$$

LWE Correctness

Recall:

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod q$$

$$\mathbf{c}^T = \mathbf{r}^T A + \mathbf{f}^T \bmod q$$

$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod q$$

$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q$$

Then

$$c' = \mathbf{r}^T(A\mathbf{s} + \mathbf{e}) + x + b\lfloor q/2 \rfloor = \mathbf{r}^T A\mathbf{s} + \mathbf{r} \cdot \mathbf{e} + x + b\lfloor q/2 \rfloor \bmod q$$

and

$$\mathbf{c} \cdot \mathbf{s} = (\mathbf{r}^T A + \mathbf{f}^T)\mathbf{s} = \mathbf{r}^T A\mathbf{s} + \mathbf{f} \cdot \mathbf{s} \bmod q$$

so

$$\begin{aligned} k &= (c' - \mathbf{c} \cdot \mathbf{s}) \bmod q \\ &= b\lfloor q/2 \rfloor + (x + \mathbf{r} \cdot \mathbf{e} - \mathbf{f} \cdot \mathbf{s}) \bmod q \end{aligned}$$

All quantities in the parentheses in the last line are "small," so k will either be close to 0 or to q/2, and will be decoded correctly.

LWE Example: Gen

Let $q = 101$, $m = n = 3$.

Suppose Bob picks

$$A = \begin{pmatrix} 63 & 52 & 64 \\ 17 & 37 & 86 \\ 38 & 28 & 25 \end{pmatrix} \quad \mathbf{s} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} \quad \mathbf{e} = \begin{pmatrix} -2 \\ -2 \\ 0 \end{pmatrix}$$

Then

$$\mathbf{w} = A\mathbf{s} + \mathbf{e} \bmod 101 = \begin{pmatrix} 63 & 52 & 64 \\ 17 & 37 & 86 \\ 38 & 28 & 25 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} + \begin{pmatrix} -2 \\ -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 25 \\ 69 \\ 50 \end{pmatrix}$$

Public key:

$$A = \begin{pmatrix} 63 & 52 & 64 \\ 17 & 37 & 86 \\ 38 & 28 & 25 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 25 \\ 69 \\ 50 \end{pmatrix}$$

Private key:

$$\mathbf{s} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

LWE Example: Enc

Public key: $A = \begin{pmatrix} 63 & 52 & 64 \\ 17 & 37 & 86 \\ 38 & 28 & 25 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 25 \\ 69 \\ 50 \end{pmatrix}$

Message bit: b=1

Alice chooses $\mathbf{r} = \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} \quad x = 1$

Ciphertext: $(\mathbf{c}, c')$, with

$$\mathbf{c} = A\mathbf{r} + \mathbf{f} \bmod 101 = \begin{pmatrix} 63 & 17 & 38 \\ 52 & 37 & 28 \\ 64 & 86 & 25 \end{pmatrix} \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 28 \\ 57 \\ 84 \end{pmatrix}$$

$$c' = \mathbf{r} \cdot \mathbf{w} + x + b\lfloor q/2 \rfloor \bmod 101 = \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 25 \\ 69 \\ 50 \end{pmatrix} + 51 = 13$$

This class is being recorded

LWE Example: Dec

Ciphertext: $\mathbf{c} = \begin{pmatrix} 28 \\ 57 \\ 84 \end{pmatrix}, c' = 13$

Private key: $\mathbf{s} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$

Bob calculates

$$k = (c' - \mathbf{c} \cdot \mathbf{s}) \bmod 101 = 13 - \begin{pmatrix} 28 \\ 57 \\ 84 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = 13 - 164 = 51$$

This is close to 50 and far from 0, so Bob decodes the message b=1.

LWE Security

This LWE-based encryption system is CPA-secure provided the LWE problem is hard. The LWE problem is in turn equivalent to the worst-case hardness of some of the lattice problems we discussed earlier, but not unfortunately for the NP-hard choices of lattice parameters.

Still, we don't know how to break the LWE assumption or protocol even with a quantum computer.

Note: This is a case where the average-case hardness that shows up in the cryptosystem is equivalent to worst-case hardness of a cleaner problem. This is a relatively strong security "proof."

Crystals-Kyber uses LWE with coefficients over rings instead of mod q.

Post-Quantum Signatures

Lattice problems and LWE can also be used to create digital signature protocols secure against known quantum algorithms.

An alternative approach uses essentially hash functions and Merkle trees.

Selected in the NIST contest:

> Crystals-Dilithium and FALCON are two lattice-based signature schemes.
>
> SPHINCS$^+$ is a hash-function-based signature scheme.