Post-Quantum Cryptography: an Overview

Andrea Sanguineti

PhD Seminars
University of Genoa, DIMA

4 May 2023

"Classical" Cryptography - An Example

- Let's consider an ubiquitous "classical" cryptosystem: RSA (Rivest-Shamir-Adleman).

  Bob's Key Generation:
- Bob selects two different large primes p and q and calculates $n := p \cdot q$.
- He calculates $\phi(n) = (p-1)(q-1)$ (Euler's totient function).
- He selects an integer e such that $\gcd(\phi(n), e) = 1$ and $1 < e < \phi(n)$.
- He calculates d such that $d \cdot e = 1 \bmod \phi(n)$ via the Euclidean algorithm.
- He publishes (e, n) as the public key and keeps (d, p, q) as the private key.

"Classical" Cryptography - An Example

- Suppose Alice wants to send a message (plaintext) $M < n$ to Bob.
- To encrypt it, she uses Bob's public key e and calculates $C := M^e \bmod n$ (ciphertext).
- Then, Alice sends C to Bob.
- Bob can recover M decrypting C with his private key d: $C^d = (M^e)^d = M \bmod n$.
- Bob's decryption is successful thanks to Fermat's little theorem and the Chinese remainder theorem.

Why do we need Post-Quantum Cryptography?

- RSA's security is based on the computational hardness of finding the factorization of n (namely p and p).
- In particular, we don't know how to factor n in polynomial time with a classical algorithm.
- But...
- ..there exists a quantum computer algorithm that allows us to calculate p and q in polynomial time!

Shor's Algorithm (1994)

It uses quantum mechanics postulates such as entanglement and measure theory.

Post-Quantum Cryptography

- A solution to this problem is finding new cryptosystems that aren't affected by the introduction of quantum algorithms.

  $\longrightarrow$ Post-Quantum Cryptography.
- It can be divided in five main areas:
- Lattice-Based Cryptography (CRYSTALS-Dilithium, CRYSTALS-Kyber, FALCON, NTRU).
- Isogeny-Based Cryptography (SIKE (broken), SIDH (broken), SQISign).
- Code-Based Cryptography (Classic McEliece, BIKE).
- Multivariate Cryptography (Unbalanced oil and vinegar, Rainbow (broken)).
- Hash-Based Cryptography (SPHINCS+).

Lattice-Based Cryptography

- Some of the most promising cryptosystems that are thought to be unvulnerable against quantum attacks are in this category, and have already been selected by NIST.
- Lattice-Based Cryptography is based on the hardness of solving problems within the context of lattices.

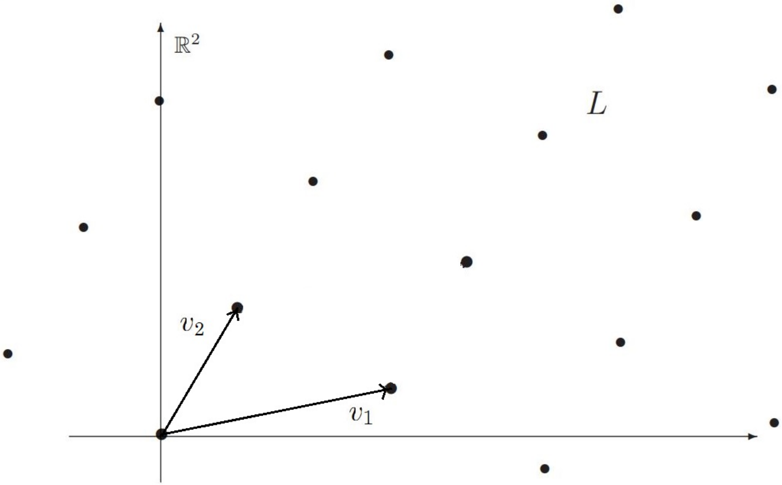
Definition (Lattice)

A lattice L in $\mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$ of full rank, or, equivalently, is the set of linear combinations with coefficients in $\mathbb{Z}$ of a basis of $\mathbb{R}^n$ $v_1, \ldots, v_n$:

$$L := \left\{ \sum_{i=1}^{n} a_i v_i \mid a_1, \ldots, a_n \in \mathbb{Z} \right\}.$$

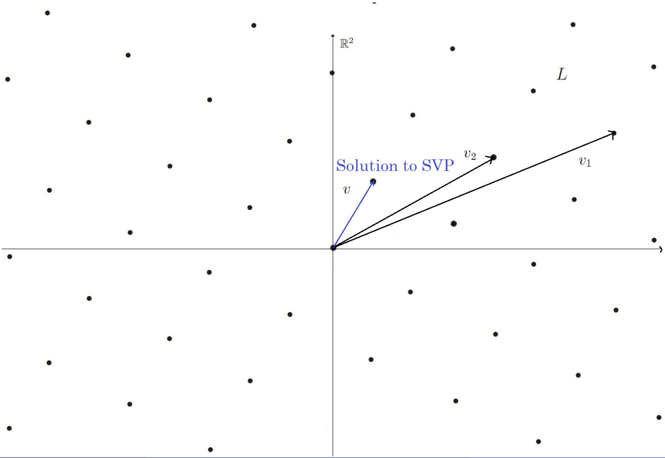
- $v_1, \ldots, v_n$ is the basis for the lattice L.

A Lattice in $\mathbb{R}^2$ with basis v_1, v_2



Definition (Shortest Vector Problem)

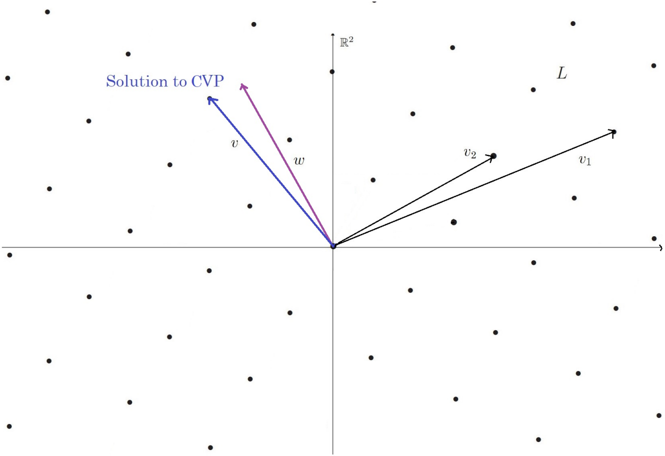
Given a Lattice L in a basis $v_1, \ldots, v_n$ find its shortest non-zero vector, namely $0 \neq v \in L$ such that $||v||$ is the smallest possible.



CVP

Definition (Closest Vector Problem)

Given a Lattice L in a basis $v_1, \ldots, v_n$ and a vector $w \in \mathbb{R}^n$, find the vector in L closest to w.



CRYSTRALS-Dilithium

- CRYSTALS-Dilithium is a NIST-standardized (2022) signature scheme based on the hardness of these lattice problems.

- It was developed by Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehlé in 2017.

CRYSTRALS-Dilithium (No pk Compression)

Key generation, S

1: $A \leftarrow M_{k,l}(R_q)$;
2: $(s_1, s_2) \leftarrow R_{q(\eta)}^{\,l} \times R_{q(\eta)}^{\,k}$;
3: $t := As_1 + s_2$;
4: $pk := (A, t), sk := (s_1, s_2)$.

$$R_q = \left\{ f = \sum_{i=0}^{n-1} a_i X^i \;\middle|\; a_i \in \mathbb{Z}_q \; \forall i = 0, \ldots, n-1 \right\}$$

$$R_{q(\gamma)} := \{ f \in R_q \mid ||f||_\infty \leq \gamma \}$$

Signing, S, sk, m

1: $y \leftarrow R_{q(\gamma_1 - 1)}^{\,l}$;
2: $w_1 := \mathrm{HighBits}(Ay, 2\gamma_2)$;
3: $c := H(m, w_1)$;
4: $z := y + c \cdot s_1$;
5: **if** $||z||_\infty \geq \gamma_1 - \beta$ or $||\mathrm{LowBits}(Ay - c \cdot s_2, 2\gamma_2)||_\infty \geq \gamma_2 - \beta$ **then** ;
6: **restart** from 1;
7: **endif** ;
8: **return** $\sigma := (z, c)$.

Verifying, V, pk, m, σ

1: $w_1' := \mathrm{HighBits}(Az - c \cdot t, 2\gamma_2)$;
2: **if** $||z||_\infty \geq \gamma_1 - \beta$ and $c == H(m, w_1')$ **then** ;
3: **accept**;
4: **else** ;
5: **reject**.

Isogeny-Based Cryptography

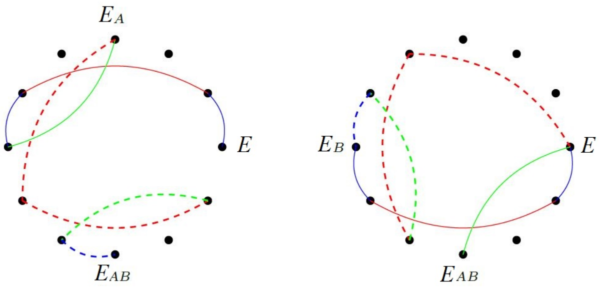
- Although some cryptosystems in this category have been broken (SIKE, SIDH) the underlying difficult problem is still hard and, to me, is the most fascinating.
- This hard problem consist in finding a path of isogenies of certain degrees between two known elliptic curves.

Definition (Isogeny)

Given two elliptic curves E_1, E_2 over a field K an isogeny over K between E_1 and E_2 is a surjective morphism of curves $\varphi : E_1 \longrightarrow E_2$ which is also a group homomorphism.

- The basic example of these types of crystosystems is the Rostovtsev-Stolbunov key-exchange (2006) (which is the Diffie-Hellman key-exchange for elliptic curves).

Rostovtsev-Stolbunov key-exchange



Public parameters	An elliptic curve E on a finite field $\mathbb{F}_q$,	
	D_π, the discriminant of the Frobenius endomorphism of E,	
	A set of primes $L = \{l_1, \ldots, l_m\}$ such that $\left(\frac{D_\pi}{l_i}\right) = 1$,	
	An eigenvalue of the Frobenius λ_i for all l_i	
Protocol	**Alice**	**Bob**
Selection of the secret path	$\rho_A \in L^*$	$\rho_B \in L^*$
Computation of the public curve	$E_A = \rho_A(E)$	$E_B = \rho_B(E)$
Exchange of the curves	$E_A \rightarrow \leftarrow E_B$	
Computation of the common secret curve	$E_{AB} = \rho_A(E_B)$	$E_{AB} = \rho_B(E_A)$

Rostovtsev-Stolbunov key-exchange and SIDH

- Even if other systems have been successfully attacked, this key exchange is still secure.
- However, it is not used because its running time is in the order of seconds, so it is not practical for everyday use.
- SIDH, which stands for Supersingular Isogeny Diffie-Hellman, is an evoultion of this scheme, which relies on more interesting mathematical structure (Isogeny Volcanoes! (see Silvia Sconza Master Thesis)).
- Unfortunately, SIDH was broken in 2022 by an attack of Wouter Castryck and Thomas Decru, which was generalized by Luciano Maino and Chloe Martindale.

- Some other NIST candidates are in this category.
- These cryptosystems are based on the hardness of decoding a general linear code.

Definition

A linear code of length n and dimension k is a linear subspace C with dimension k of the vector space $\mathbb{F}_q^n$ where $\mathbb{F}_q$ is the finite field with q elements.

- An example of such a scheme is the McEliece Cryptosystem.

Key generation:

- The principle is that Alice chooses a linear code C from some family of codes for which she knows an efficient decoding algorithm, and to make C public knowledge but keep the decoding algorithm secret.
- Alice selects a binary (n, k) linear code C capable of (efficiently) correcting t errors from some large family of codes.
- This choice should give rise to an efficient decoding algorithm A.
- Let also G be any generator matrix for C.
- Alice selects a random $k \times k$ binary invertible matrix S.
- Alice selects a random $n \times n$ permutation matrix P.
- Alice computes the $k \times n$ matrix $\hat{G} = SGP$.
- Alice's public key is $(\hat{G}, t)$. The secret key is (S, P, A).

Message Encryption:

- Bob wants to send a message m to Alice, whose public key is $(\hat{G}, t)$.
- Bob encodes the message m as a binary string of length k.
- Bob computes the vector $c' = m\hat{G}$.
- Bob generates a random n bit vector z containing exactly t ones.
- Bob computes the ciphertext as $c = c' + z$, and sends it to Alice.

Message Decryption:

- After having received c, Alice decrypts the message in the folloing way.
- Alice computes the inverses of P and $S \cdot G$ (i.e P^{-1} and the right inverse $(S \cdot G)^{-1}$).
- Alice computes $\hat{c} = cP^{-1}$.
- Alice uses the decoding algorithm A to decode $\hat{c}$ to $\hat{m}$.
- Alice computes $m = \hat{m}(S \cdot G)^{-1}$.

Code-Based NIST Round 4 Submissions

The Code-Based NIST Round 4 Submissions are:

- BIKE (Bit Flipping Key Encapsulation).
- Classic McEliece (which is based on the algorithm presented before).
- HQC (Hamming Quasi-Cyclic).

Multivariate Cryptography

- The hardness of these cryptosystems is based on the hardness of the following problem.
- Given a finite field of q elements $\mathbb{F}_q$ and m quadratic polynomials $p_1, \ldots, p_m \in \mathbb{F}_q[X_1, \ldots, X_n]$ in n variables, find a solution $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ of the system of equations

$$p_i(X_1, \ldots, X_n) = 0, \; i = 1, \ldots, m.$$

In a multivariate public key cryptosystem we have the following.

- The public key $pk = (p_1, \ldots, p_m)$ consists of a m-tuple of quadratic polynomials $p_1, \ldots, p_m \in \mathbb{F}_q[X_1, \ldots, X_n]$ in n variables.
- The encryption function is the polynomial map $E : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^m$ defined by

$$E(X_1, \ldots, X_n) = (p_1(X_1, \ldots, X_n), \ldots, p_m(X_1, \ldots, X_n)).$$

- The secret key consists of data on how $p_1, \ldots, p_m$ have been generated (it depends on the cryptosystem) and makes possible to easily invert E using the decryption function.

Multivariate Cryptography

- Direct attacks to these types of schemes mainly employ the calculation of Gröebner basis.
- The easisest algorithm to compute them is Buchberger's algorithm, but it is also the slowest.
- Many improved methods have been proposed, in particular the F_4 and F_5 algorithms, due to Faugére, and their many variations.

The SIKE Song

Eurocrypt 2023 - Lyon, France - Rump Session
https://youtu.be/b_AuzlaIxLs
Credits: The Isogeny Club.

Thanks for your attention!