Post Quantum Cryptography

Eleena Gupta

Indian Institute of Science, Bangalore

21 April 2023

Cryptology

- Cryptography: foundation of security for electronic transaction.
  - Primary Goal: Confidentiality and Authenticity
- Cryptanalysis: the art and science of breaking cryptosystems.
  - Played a pivotal role in the development of Computer Science and Technology.

RSA Signature Scheme

- **Key-Gen:** Chose two *large* primes p and q.
  - Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$.
  - Randomly choose an odd number e that is co-prime to $\phi(n)$.
  - Compute d s.t. $ed \equiv 1 \bmod \phi(n)$.
  - Pick a "cryptographic" hash function $H : \{0,1\}^* \to [1, n-1]$.
  - Public key: (e, n, H); secret signing key: d.

RSA Signature Scheme

- **Key-Gen:** Chose two *large* primes p and q.
  - Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$.
  - Randomly choose an odd number e that is co-prime to $\phi(n)$.
  - Compute d s.t. $ed \equiv 1 \bmod \phi(n)$.
  - Pick a "cryptographic" hash function $H : \{0,1\}^* \to [1, n-1]$.
  - Public key: (e, n, H); secret signing key: d.
- **Sign:** Given a message $m \in \{0,1\}^*$ compute $H(m)$ and the signature $\sigma = H(m)^d \bmod n$.
- **Verify:** Given (m, σ) compute $H(m)$ and $\sigma^e \bmod n$. Accept σ as a valid signature on m if and only if $H(m) = \sigma^e \bmod n$.

Hard Problem

Factorisation: Given the RSA modulous n, find its prime factors.
If one can solve the Factorisation problem then one can

1. Easily forge a signature.
2. Obtain the session key k from a ciphertext c.

So how difficult is Factorisation?

Real World Crypto

- For actual deployment a cryptosystem needs to satisfy very stringent security and efficiency criteria.
- Ideal: Breaking the cryptosystem is as hard as solving some well-studied (number theoretic) problem.
  - factorisation or finding discrete log of a 4096-bit number, inverting the AES-128 function, finding collision in SHA-256...
  - Based on our current understanding, with appropriate parameter choices these problems appear to be hard.

What's Your Model of Computation?

- Any physically computable function can be computed on a Turing Machine with at most polynomial increase in the running time.
- If we are interested in which problems can be solved efficiently on a realistic model of computation, we can restrict attention to a probabilistic Turing Machine.
- It appears that a Turing Machine takes exponential overhead to simulate systems at the sub-atomic level.
    - Turing Machine follows classical laws of physics.
- Why not try to build a computer based on quantum mechanics which is the theory for sub-atomic physics?

Quantum Computer

- Conceived independently by Yuri Manin (1980) and Richard Feynman (1981).
- Use quantum-mechanical phenomena such as superposition to perform operations on data.
- qubit: quantum analogue of classical bit, and can be in two states at the same time, each with a certain probability.
- An n-qubit register can be in 2^n states at the same time, each with a certain probability.
- When measured, the register reverts to being in one of the 2^n states according to its probability distribution.

Quantum Algorithm

- 1985: David Deutsch developed the idea of Quantum Turing Machine.
  - Asked whether quantum computers can be useful for classical problems.
  - Showed a single query suffices to decide whether a one-bit function is constant or balanced.
- 1994: Peter Shor proposed a quantum algorithm for factorisation in polynomial time. Solves DLP as well.
- 1997: Lov Grover developed a quantum search algorithm with $\approx \sqrt{N}$ complexity, where N is the size of the unsorted database.
  - AES-128 key can be recovered in $\approx 2^{64}$ operations.

Post-Quantum Cryptography

- CESG Whitepaper concludes:
  By contrast, post-quantum public key cryptography appears to offer much more effective mitigations for real-world communications systems from the threat of future quantum computers.
- Post-quantum Cryptosystems: run on conventional/classical computers but are secure against attacks by quantum computers.

Potential Candidates

1. Information-Theoretic Security: One-Time Pad (1882).
2. Symmetric-key Cryptography: Advanced Encryption Standard or AES (1998).
3. Hash-Based Cryptography: Merkle's hash-tree Public Key Signature (1979).
4. Multivariate-Quadratic Based Cryptography: Patarin's Signature Scheme (1996).
5. Code-Based Cryptography: McEliece's Public Key Encryption (1978) based on Hidden-Goppa-Code.
6. Lattice-Based Cryptography: "NTRU" PKE by Hoffstein, Pipher and Silverman (1998).
7. Isogeny-Based Cryptography: Supersingular Isogeny Diffie-Hellman Key Exchange by Feo, Jao and Plut (2011).

Code-Based Crypto

Coding Theory

- Primary concern of Coding Theory is efficient and reliable transmission/storage of data in the presence of (random) noise.
- The essential idea is to add redundancy.
  - k-bit data is expanded to, say, n-bit so that errors can be detected and corrected.
- The aim is to construct efficient encoding/decoding techniques to correct as many errors as possible without adding too much redundancy.
- Code-Based Crypto: Apply techniques of Coding Theory in the construction of cryptographic schemes.

Linear Codes

A binary linear code $\mathbb{C}$ of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_2^n$.

- Here, all messages are k-bits to which we add $(n-k)$ additional (redundant) bits to get codewords of length n-bits.
- Let $m = \langle 1001 \rangle$: one simple strategy is to repeat all the bits once: $c = \langle 10011001 \rangle$.
- This is called replication code.
- Suppose, one error has occured; then we can detect but can we correct it?

Linear Codes

A binary linear code $\mathbb{C}$ of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_2^n$.

- Here, all messages are k-bits to which we add $(n-k)$ additional (redundant) bits to get codewords of length n-bits.
- Let $m = \langle 1001 \rangle$: one simple strategy is to repeat all the bits once: $c = \langle 10011001 \rangle$.
- This is called replication code.
- Suppose, one error has occured; then we can detect but can we correct it?
- We need some "clever" strategy for error correction.
- Suppose, given $m = \langle a \ b \ c \ d \rangle$ we form the corresponding code as

$$c = \langle a \ b \ c \ d \ a+b \ c+d \ a+d \ b+c \rangle$$

- Here, the last 4 are parity bits.

Generator Matrix

The encoding strategy of previous slide can be expressed as a matrix called the Generator Matrix.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

So, $\mathbb{C}$ is the row space of the generator matrix $G \in \mathbb{F}_2^{k \times n}$:

$$\mathbb{C} = \{\mathbf{m} \cdot G : m \in \mathbb{F}_2^k\}$$

Hamming Weight and Distance

- The Hamming weight of a word is the number of nonzero coordinates in that word.

$$\mathrm{wt}(100101) = 3$$

- The Hamming distance between two n-tuples over Σ is the number of coordinate positions where they differ.

$$\mathrm{d}(100101, 110100) = 2$$

- For $\Sigma = \{0,1\}$, the Hamming distance between $x, y \in \{0,1\}^n$ is same as Hamming weight of $x + y$.
- The Hamming distance of a code $\mathbb{C}$ is

$$\mathrm{d}(\mathbb{C}) = \min\{\mathrm{d}(x,y) : x, y \in \mathbb{C} \text{ and } x \neq y\}$$

Minimum Distance

The minimum distance of a code $\mathbb{C}$:

$$\begin{aligned} d(\mathbb{C}) &= \min\{\mathrm{d}(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathbb{C} \text{ and } \mathbf{x} \neq \mathbf{y}\} \\ &= \min\{\mathrm{wt}(\mathbf{w}) : \mathbf{w} \in \mathbb{C} \text{ and } \mathbf{w} \neq \mathbf{0}\} \end{aligned}$$

So, the codewords of $\mathbb{C}$ can be visualised as points in some space that are at least $d(\mathbb{C})$ distance away from each other.

Decoding Problem

- You are given $\mathbf{w} \in \mathbb{F}_2^n$. Let $\mathbf{c} \in \mathbb{C}$ be a unique closest codeword. Your task is to find $\mathbf{c}$
- Let $\mathbf{w} = \mathbf{c} + \mathbf{e}$: an equivalent problem is to find $\mathbf{e}$.
- Suppose the Hamming weight of $\mathbf{e}$ is t, then this is the t-error correcting problem.

Decoding Problem

- You are given $\mathbf{w} \in \mathbb{F}_2^n$. Let $\mathbf{c} \in \mathbb{C}$ be a unique closest codeword. Your task is to find $\mathbf{c}$
- Let $\mathbf{w} = \mathbf{c} + \mathbf{e}$: an equivalent problem is to find $\mathbf{e}$.
- Suppose the Hamming weight of $\mathbf{e}$ is t, then this is the t-error correcting problem.
- For a code $\mathbb{C}$ with minimum distance $d = 2t + 1$, any vector $\mathbf{w} = \mathbf{c} + \mathbf{e}$ such that $\mathrm{wt}(\mathbf{e}) \leq t$ can be uniquely decoded to $\mathbf{c}$.
  - Imagine a sphere of radious t with $\mathbf{c}$ as centre – clearly there is no closer code word.
  - Nearest Neigbor Decoding – in the worst case one needs to compare $\mathbf{w}$ with all the 2^k codewords in $\mathbb{C}$.
- If there are more errors, say upto $2t$: may be detected but cannot be corrected.

Decoding is a Hard Problem

- Decoding is hard for random codes: if the linear expansion is random then decoding is NP-complete.
  - A general decoding strategy called Information-Set Decoding takes exponential time.
- Information Set: a set of coordinates that can uniquely determine a codeword.
  - Information-Set Decoding: randomly select a subset of coordinates in the received word and assume there is no error. Then try to solve for the transmitted message.
- A primary concern for Coding Theory is design of good codes having fast decoding algorithms:
  - Reed-Solomon codes, Goppa codes, BCH codes, ...

Encoding as Encryption

- Suppose $\mathbb{C}$ is our $[n, k, t]$ code with generator matrix $G \in \mathbb{F}_2^{k \times n}$.
- Assume that there is an efficient decoding algo for $\mathbb{C}$.
- Let $\mathbf{m} \in \{0, 1\}^k$.
- Pick an error-vector $\mathbf{e}$ with $\mathrm{wt}(e) = t$.
- Construct the ciphertext as

$$\mathbf{y} = \mathbf{m}G + e$$

Decryption?
What about security?
Completely insecure!

Hiding the Code Structure

- Encoding technique needs to be public – anybody should be able to encrypt.
  - Decoding is easy if you have the corresponding private key.
  - But, must be hard just based on the public key.
- The code should look like random unless you know the trapdoor!
- This is the primary design goal in Code-Based Public Key Encryption.

McEliece Encrypton Scheme [1978]

- $\mathbb{C}$: Some "suitable" code of length n and dimension k having minimum distance $2t+1$.
- G: a generator matrix for $\mathbb{C}$ for which there is a fast decoding algo with t error-correcting capability.

McEliece Encrypton Scheme [1978]

- $\mathbb{C}$: Some "suitable" code of length n and dimension k having minimum distance $2t + 1$.
- G: a generator matrix for $\mathbb{C}$ for which there is a fast decoding algo with t error-correcting capability.
- Pick a random non-singular matrix $S \in \mathbb{F}_2^{k \times k}$
- Pick a random permutation matrix $P \in \mathbb{F}_2^{n \times n}$
- S and P are used to randomly shuffle and permute G:

$$G' = S \cdot G \cdot P$$

McEliece Encryption

- Public key: (n, k, t) and the $k \times n$ matrix G'.
- Secret key: (S, P) and a fast decoding algo for G.
- Encrypt: Given $\mathbf{m} \in \{0,1\}^k$, choose $\mathbf{e} \in_R \mathbb{F}_2^n$ with $\mathrm{wt}(e) = t$ and compute the ciphertext

$$\mathbf{y} = \mathbf{m} \cdot G' + e$$

McEliece Decryption

- Compute:

$$\mathbf{y} \cdot P^{-1} = \mathbf{m} \cdot G' \cdot P^{-1} + \mathbf{e} \cdot P^{-1} = (\mathbf{m} \cdot S) \cdot G + \mathbf{e} \cdot P^{-1}$$

- P is a permutation matrix so $\mathrm{wt}(\mathbf{e} \cdot P^{-1}) = \mathrm{wt}(\mathbf{e})$
- Now use the fast decoding for $\mathbb{C}$ to compute $\mathbf{m} \cdot S$ and then $\mathbf{m}$.
- What is the problem for an adversary?

McEliece Decryption

- Compute:

$$\mathbf{y} \cdot P^{-1} = \mathbf{m} \cdot G' \cdot P^{-1} + \mathbf{e} \cdot P^{-1} = (\mathbf{m} \cdot S) \cdot G + \mathbf{e} \cdot P^{-1}$$

- P is a permutation matrix so $\mathrm{wt}(\mathbf{e} \cdot P^{-1}) = \mathrm{wt}(\mathbf{e})$
- Now use the fast decoding for $\mathbb{C}$ to compute $\mathbf{m} \cdot S$ and then $\mathbf{m}$.
- What is the problem for an adversary?
  - Decode $\mathbf{y}$ to its nearest codeword $\mathbf{m} \cdot G'$.
  - Assuming G' is random this is precisely the general decoding problem.

One Way Encryption

- McEliece scheme is a one way encryption.
- Insecure if the adversary is given access to a decryption oracle.
  - Given $\mathbf{y}$, adversary $\mathcal{A}$ can ask for the decryption of any $\mathbf{y}' \neq \mathbf{y}$.
- $\mathcal{A}$ prepares a codeword $\mathbf{c}' = \mathbf{m}' \cdot G'$ and asks for decryption of $\mathbf{y}' = \mathbf{y} + \mathbf{c}'$.
- Oracle returns $\mathbf{m} + \mathbf{m}'$ from which $\mathcal{A}$ extracts $\mathbf{m}$.
- Use McEliece as a Key Encapsualtion Mechamnism (KEM):
  - Choose random $\mathbf{e}$ with weight t and use $H(e)$ as the key for some secure symmteric key encryption technique.
  - Some authentication mechanism is also incorporated to achieve security against chosen-ciphertext attack.

Instantiation

- McEliece suggested using binary Goppa Code with $n = 1024$, $k = 524$ and $t = 50$.
  - The same code is used today but parameter size has been increased significantly based on the cryptanalytic efforts.
- Several other codes have been suggested for code-based crypto:
  - Reed-Muller codes, concatenated codes, cyclic codes . . .
  - Most of these proposals have been subsequently broken.
- McEliece using binary Goppa Code is a potential candidate for quantum-safe crypto.
- Classic McEliece is a candidate in NIST Round 4 PQC Standardization Competition.