Fully Homomorphic Encryption &

Post quantum Cryptography

Today: - Post quantum cryptography
- * LWE assumption

- Fully Homomorphic Encryption (FHE)
  - * Definitions
  - * Applications
  - * Construction

Post Quantum Cryptography

- All the assumptions that we have seen so far can be broken with a quantum computer Including: RSA, Discrete Log in $\mathbb{Z}_p^*$ & in elliptic curves.

Will modern cryptography die with the birth of quantum computers?

Hopefully not..

There are assumptions that are believed to resist quantum attacks, and we know how to build crypto from such assumptions.

Learning with Error (LWE) Assumption

There is a family of "lattice-based" assumptions that are not known to be broken using a quantum computer, and are usefull in cryptograpy.

The most commonly used is the LWE assumption, introduced by Oded Regev in 2004.

LWE Assumption: It is hard to solve noisy linear equations (recall that it is easy to solve linear equations without noise by Gaussian Elimination)

Formally: LWE is associated with parameters:

$$\overset{\text{Field size}}{q}\ ,\ \overset{\text{\# of var's}}{n}\ ,\ \overset{\text{\# of eqn's}}{m}\ ,\ \text{error dist. } \chi_q$$

$$(q \text{ prime} \quad m \gg n)$$

For random $s \leftarrow \mathbb{Z}_q^n$

random $a_1, \dots a_m \leftarrow \mathbb{Z}_q^n$

$e_1, \dots, e_m \leftarrow \chi_q$

$$\text{Given} \quad \begin{array}{l} a_1,\ a_1 s + e_1 \\ \vdots \\ a_m,\ a_m s + e_m \end{array} \xrightarrow{\text{HARD}} s$$

Decisional LWE (DLWE)

$$\begin{pmatrix} a_1, s a_1 + e_1 \\ \vdots \\ a_m, s a_m + e_m \end{pmatrix} \cong \begin{pmatrix} a_1, u_1 \\ \vdots \\ a_m, u_m \end{pmatrix}$$

where $u_1, \dots, u_m \leftarrow \mathbb{Z}_q$

Matrix notation : $(A, sA + E) \cong (A, U)$

where $A \leftarrow \mathbb{Z}_q^{n \times m}$ $s \leftarrow \mathbb{Z}_q^n$ $E \leftarrow \chi_q^{n \times m}$ $U \leftarrow \mathbb{Z}_q^m$

- We do not know how to break this assumption with quantum computers (as opposed to Factoring & DL).

- No known sub-exp. algorithms.

- Reduces to worst case lattice assumptions.

- Resilient to leakage.

- We can construct public key encryption, digital signatures, collision resistant hash functions, identity based encryption, fully homorphic encryption ... from Decisional LWE.

Not yet used in practice because less efficient (keys are huge) & because we do not have quantum computers.

- A few years ago NIST solicited for quantum resilient public-key cryptographic primitives.

2018: First post-quantum Cryptography Standardization Confrence.

Fully Homomorphic Encryption (FHE) from DLWE.

A notion suggested by Rivest-Adleman-Dertouzos 78.

$$\mathrm{Enc}(PK, b_1), \quad \mathrm{Enc}(PK, b_2)$$

$$\downarrow$$

$$\mathrm{Enc}(PK, b_1+b_2), \quad \mathrm{Enc}(PK, b_1 \cdot b_2)$$

- Did not provide a construction.

- First construction: Gentry 2009 (lattice-based)

2011: Brakerski & Vaikuntanathan constructed FHE from DLWE.

Note : El-Gamal & (Vanila) RSA are homomorphic w.r.t. mult. but not addition.

RSA: $m_1^e \bmod n$, $m_2^e \bmod n \longrightarrow (m \cdot m_2)^e \bmod n$

El-Gamal: $g^{r_1}, g^{xr_1} \cdot m_1$, $g^{r_2}, g^{xr_2} \cdot m_2 \longrightarrow g^{r_1+r_2}, g^{x(r_1+r_2)} \cdot m_1 m_2$

Applications:

- Private delegation : A user can delegate all her private data to the cloud using FHE.

  The cloud can perform computations on the encrypted data blindly , without learning any information.

- Secure computation with minimal communication.

  ⋮

Construction [Gentry - Sahai - Wichs 2013]

KeyGen(1^n): $A \leftarrow \mathbb{Z}_q^{(n-1)\times m}$ q - prime
can be of size poly(n)

$s \leftarrow \mathbb{Z}_q^{n-1}$ $m > n \cdot \log q$.
$(m = \Theta(n \log q))$

$e \leftarrow \chi_q^m$

$$PK = B = \begin{pmatrix} A \\ sA+e \end{pmatrix} \qquad SK = t = (-s, 1) \in \mathbb{Z}_q^n$$

Note: $t \cdot B \approx 0$

Encrypt(b): Uses special fixed matrix
$G \in \mathbb{Z}_q^{n\times N}$ (where $N = n \cdot (\lfloor \log q \rfloor + 1)$)

$$\mathrm{Enc}(\underset{\underset{B}{\shortparallel}}{pk}, b) = B \cdot R + bG \in \mathbb{Z}_q^{n\times N}$$

for randomly chosen $R \leftarrow \{0,1\}^{m\times N}$

$\mathrm{Dec}(\overset{\overset{t=(-s,1)}{\shortparallel}}{sk}, C)$: Compute $\underset{\substack{\cap \\ \mathbb{Z}_q^{1\times n}}}{t} \cdot \underset{\substack{\cap \\ \mathbb{Z}_q^{n\times N}}}{C}$

It $t \cdot C \approx 0$ output $b = 0$, and o.w. output $b = 1$.

Correctness: $t \cdot (BR + bG) = \overbrace{(t\cdot B)\cdot R}^{\approx 0} + b\,\overbrace{tG}^{\text{large.}}$ (property of G)

Semantic Security: Follows from DLWE assumption:

If B was uniform $B \leftarrow \mathbb{Z}_q^{n \times m}$ then $B \cdot R$ for

$R \leftarrow \{0,1\}^{m \times N}$ would have been truly random

[Follows from the Leftover-Hash-Lemma & from the fact that $m > n \cdot \log q$.]

-Impagliazzo-Levin-Luby 89

Thus, $\underset{\underset{\mathrm{Enc}(PK,b)}{\shortparallel}}{BR + bG}$ would hide b information theoretically.

Since by DLWE $B \cong U$ it follows that

$\mathrm{Enc}(B,b)$ must also hide b for B generated

by KeyGen.

Homomorphic Operations:

Addition: $C_1 = BR_1 + b_1 G$

$$C_2 = BR_2 + b_2 G$$

$$C^+ = C_1 + C_2 = B\underbrace{(R_1 + R_2)}_{\substack{\text{small} \\ \text{(but slowly grows...)}}} + \underbrace{(b_1 + b_2)}_{\text{addition in } \mathbb{Z}_q.} G$$

- To get addition mod 2:

We will soon see how to compute $C^{\times}$ which is an encryption of $b_1 \cdot b_2$ (mod 2).

Can use this to get addition mod 2 by homomorphical computing $(b_1 + b_2) - 2b_1b_2$ mod q

- The growth of R:

This is indeed a problem (especially for $C^{\times}$ where R grows even more)

indeed this scheme provides only leveled homomorphism (which means that we can only homomorphic do bouded depth computations).

There is a beautiful technique called bootstrapping to get around this issue.

Multiplication :

To homomorphically multiply two encrypted bits need to use an additional property of the matrix G. (beyond $t \cdot G$ being large which is needed for correctness)

$G \in \mathbb{Z}_q^{n \times N = n(\lfloor \log q \rfloor + 1)}$ has the property that

$\exists$ eff. computable function $G^{-1} : \mathbb{Z}_q^{n \times N} \to \{0,1\}^{N \times N}$

s.t. $\forall M \in \mathbb{Z}_q^{n \times N}$ $\quad G \cdot (G^{-1}(M)) = M.$

$$G = \begin{array}{c} \\ n \\ \\ \end{array} \begin{pmatrix} 1 \; 2 \ldots 2^{\lfloor \log q \rfloor} & & \\ & \ddots & \\ & & 1 \; 2 \ldots 2^{\lfloor \log q \rfloor} \end{pmatrix}$$

G^{-1} = bit decomposition function

$$
\begin{aligned}
C^{\times} &= \overset{\mathbb{Z}_q^{n\times N}}{\overset{\rotatebox{90}{$\in$}}{C_1}} \cdot \overset{\mathbb{Z}_q^{N\times N}}{\overset{\rotatebox{90}{$\in$}}{\overline{G^{-1}(C_2)}}} \\
&= (BR_1 + b_1 G) \cdot G^{-1}(C_2) \\
&= B\, R_1 \cdot G^{-1}(C_2) + b_1 \cdot (G \cdot G^{-1}(C_2)) \\
&= B \cdot (R_1 \cdot G^{-1}(C_2)) + b_1 C_2 \\
&= B \cdot (R_1\, G^{-1}(C_2)) + b_1 (BR_2 + b_2 G) \\
&= B \underbrace{(R_1 G^{-1}(C_2) + b_1 R_2)}_{\text{small}} + b_1 b_2 G
\end{aligned}
$$