Fully Homomorphic Encryption and Post Quantum Cryptography

6.857

Post Quantum Cryptography

All the assumptions that we have seen so far for public key cryptography are broken using quantum computers

Factoring, RSA, Discrete Log, Elliptic Curves…

Is Crypto Going to Die??

- There is a family of assumptions that are believed to **resist quantum attacks**.

- We know how to **build crypto-systems** from these assumptions.

  more advanced

Today

1. Define **Learning with Error** (LWE) assumption, which is believed to be post-quantum secure

2. **Fully Homomorphic Encryption** (FHE)

   - Definition
   - Application
   - Construction from LWE

Learning with Error (LWE)

[Regev 2004]

LWE assumption: It is **hard** to solve **random noisy linear equations**

Note: It is easy to solve linear equations without noise (Gaussian Elimination)

Learning with Error (LWE)

[Regev 2004]

Formally: LWE is associated with parameters (q, n, m, χ)

q = field size (prime)
n = # variables
m = # equations ($m \gg n$)
χ = error distribution

Decisional version

$\boldsymbol{LWE_{q,n,m,\chi}}$: For random $s \leftarrow Z_q^n$, random $A \leftarrow Z_q^{n\times m}$, and $e \leftarrow \chi^m$,

$$(\boldsymbol{A}, \boldsymbol{sA} + \boldsymbol{e}) \approx (\boldsymbol{A}, \boldsymbol{U})$$

$\boldsymbol{LWE_{q,n,m,\chi}}$: For random $s \leftarrow Z_q^n$, random $A \leftarrow Z_q^{n\times m}$, and $e \leftarrow \chi^m$,

$$(\boldsymbol{A}, \boldsymbol{sA} + \boldsymbol{e}) \approx (\boldsymbol{A}, \boldsymbol{U})$$

1. Believed to resist quantum attacks.
2. No known sub-exponential algorithms.
3. Reduces to worst-case lattice assumptions
4. Resilient to leakage
5. We can construct amazing cryptographic primitives from it, such as **fully homomorphic encryption**!

Fully Homomorphic Encryption

- Notion suggested by Rivest-Adleman-Dertouzos in 1978:

$$Enc(pk, x), Enc(pk, y) \xrightarrow{\text{easy}} Enc(pk, x + y)$$

$$Enc(pk, x), Enc(pk, y) \xrightarrow{\text{easy}} Enc(pk, x \cdot y)$$

Addition and multiplication mod 2 are complete

$$Enc(pk, x) \xrightarrow{\text{easy}} Enc(pk, f(x))$$

Fully Homomorphic Encryption

- Notion suggested by Rivest-Adleman-Dertouzos in 1978:

$$Enc(pk, x), Enc(pk, y) \xrightarrow{\text{easy}} Enc(pk, x + y)$$

$$Enc(pk, x), Enc(pk, y) \xrightarrow{\text{easy}} Enc(pk, x \cdot y)$$

- **First construction** by Gentry 2007 (lattice based).
- **First construction under LWE** by Brakerski and Vaikuntanathan 2011.
- **Today:** We will see construction by Gentry-Sahai-Waters 2013

Fully Homomorphic Encryption

- Notion suggested by Rivest-Adleman-Dertouzos in 1978:

$$Enc(pk,x), Enc(pk,y) \xrightarrow{\text{easy}} Enc(pk,x+y)$$

$$Enc(pk,x), Enc(pk,y) \xrightarrow{\text{easy}} Enc(pk,x \cdot y)$$

- Note: RSA and El-Gamal are homomorphic w.r.t. **multiplication**, but not addition:

RSA: $x^e \bmod n,\ y^e \bmod n \xrightarrow{\text{easy}} (xy)^e \bmod n$

El-Gamal: $(g^{r_1}, g^{r_1 s} \cdot x),\ (g^{r_2}, g^{r_2 s} \cdot y) \xrightarrow{\text{easy}} (g^{r_1+r_2}, g^{(r_1+r_2)s} \cdot xy)$

Applications of FHE: Private Delegation

- Suppose we want to delegate our computation (say to the cloud)
- Suppose we don't want the cloud to know what the computation is.

Paradox?

Can do private delegation using FHE!

Construction

[Gentry-Sahai-Waters13]

$m = \theta(n \log q)$

$\boldsymbol{Gen(1^n)}$: $A \leftarrow Z_q^{(n-1)\times m}$
$s \leftarrow Z_q^{n-1}$
$e \leftarrow \chi^m$

$$PK = B = \begin{pmatrix} A \\ sA + e \end{pmatrix} \in Z_q^{n\times m}$$

$$SK = t = (-s, 1) \in Z_q^n$$

$tB \approx 0$

$\boldsymbol{Enc(PK, b)}$: Choose at random $\mathrm{R} \leftarrow \{0,1\}^{m\times N}$, output

$$\mathbf{CT} = \boldsymbol{BR} + \boldsymbol{bG} \in \boldsymbol{Z_q^{n\times N}},$$

$N = n(\log q + 1)$

where $G \in Z_q^{m\times N}$ is a fixed matrix

tG is large

$$G = \begin{pmatrix} 1\ 2\ 4\ \dots\ 2^{\log q} & & \\ & \ddots & \\ & & 1\ 2\ 4\ \dots\ 2^{\log q} \end{pmatrix}$$

Construction

[Gentry-Sahai-Waters13]

Gen(1^n): $A \leftarrow Z_q^{(n-1)\times m}$ ($m = \theta(n \log q)$)

$s \leftarrow Z_q^{n-1}$

$e \leftarrow \chi^m$

$$PK = B = \begin{pmatrix} A \\ sA + e \end{pmatrix} \in Z_q^{n\times m}$$

$$SK = t = (-s, 1) \in Z_q^n \qquad tB \approx 0$$

Enc(PK, b): Choose at random $R \leftarrow \{0,1\}^{m\times N}$, output

$$\mathbf{CT} = \boldsymbol{BR} + \boldsymbol{bG} \in \boldsymbol{Z_q^{n\times N}}, \qquad N = n(\log q + 1)$$

where $G \in Z_q^{m\times N}$ is a fixed matrix

Dec(SK, CT): Compute $t \cdot CT$, and output 0 iff $t \cdot CT \approx 0$.

Correctness: $\boldsymbol{R}$ **is small**, and $\boldsymbol{t} \cdot \boldsymbol{G}$ **is large**, hence:

$$t \cdot CT = t \cdot BR + btG \approx 0 + btG.$$

Construction

[Gentry-Sahai-Waters13]

Gen$(\mathbf{1^n})$: $A \leftarrow Z_q^{(n-1)\times m}$ $m = \theta(n \log q)$

$s \leftarrow Z_q^{n-1}$

$e \leftarrow \chi^m$

$$PK = B = \begin{pmatrix} A \\ sA + e \end{pmatrix} \in Z_q^{n\times m}$$

$$SK = t = (-s, 1) \in Z_q^n$$

$tB \approx 0$

Enc$(\boldsymbol{PK}, \boldsymbol{b})$: Choose at random $\mathrm{R} \leftarrow \{0,1\}^{m\times N}$, output

$$\mathbf{CT} = \boldsymbol{BR} + \boldsymbol{bG} \in \boldsymbol{Z_q^{n\times N}},$$

$N = n(\log q + 1)$

where $G \in Z_q^{m\times N}$ is a fixed matrix

Security: If B was random in $Z_q^{n\times m}$ then $(\mathrm{B}, BR) \equiv (B, U)$
(by the Leftover Hash Lemma, follows from the fact that $m > n \log q$).
⟶ **By LWE,** $(B, BR) \approx (B, U)$

Computing on Encrypted Data

$\boldsymbol{Enc(PK, b)}$: Choose at random $R \leftarrow \{0,1\}^{m\times N}$, output

$$\mathbf{CT} = \boldsymbol{BR} + \boldsymbol{bG} \in \boldsymbol{Z_q^{n\times N}},$$

$N = n(\log q + 1)$

where $G \in Z_q^{m\times N}$ is a fixed matrix

$$BR_1 + b_1G,\ BR_2 + b_2G \xrightarrow{\textbf{easy}} \boldsymbol{CT^+} = CT_1 + CT_2 = B(R_1{+}R_2) + (b_1{+}b_2)G$$

$\boldsymbol{G^{-1}}: \boldsymbol{Z_q^{n\times N}} \rightarrow \{\mathbf{0},\mathbf{1}\}^{\boldsymbol{N\times N}}$ **is bit decomposition function:** $\forall \mathbf{M} \in \boldsymbol{Z_q^{n\times N}}\ \ \boldsymbol{GG^{-1}(M) = M}$.

mod q, we want mod 2

$$CT_1,\ CT_2 \xrightarrow{\textbf{easy}} \boldsymbol{CT^\times} = CT_1 \cdot \underbrace{G^{-1}(CT_2)}_{\text{in } \{0,1\}^{N\times N}} = (BR_1 + b_1G)\cdot G^{-1}(CT_2)$$

$$= BR' + b_1 \cdot CT_2 = B(R'{+}b_1R_2) + b_1b_2G = BR'' + b_1b_2G$$

Can get addition mod 2 by computing $CT^+ - 2CT^\times$

The Error Grows!

$$\underbrace{BR_1 + b_1G}_{CT_1},\ \underbrace{BR_2 + b_2G}_{CT_2} \xrightarrow{\text{easy}} CT^+ = CT_1 + CT_2 = B(R_1{+}R_2) + (b_1{+}b_2)G$$

$$CT_1,\ CT_2 \xrightarrow{\text{easy}} CT^\times = CT_1 \cdot G^{-1}(CT_2) = (BR_1 + b_1G)\cdot G^{-1}(CT_2)$$

$$= BR' + b_1 \cdot CT_2 = B(R'{+}b_1R_2) + b_1b_2G = BR'' + b_1b_2G$$

Bootstrap to reduce the noise!


THANK
YOU