Attribute-Based Encryption

Tuesday, September 29, 2020 11:35 AM

FHE: compute over encrypted data

FH. Signature: "verify computation".

ABE: Access structure.

msk

sk_P

x, m — attribute: intents; message: contact info

P — policy) match?

m if $P(x)$ is true.

Security: if $P(x)$ is false → m hidden.

Def: ABE.

- **Setup**: mpk / master public key; msk / master secret key
- **KeyGen** (msk, P): sk_P
- **Enc** (mpk, x, m): $ct(x,m)$
- **Dec** ($ct(x,m)$, sk_P): m.

Correctness: $\mathrm{Dec}(ct(x,m), sk_P): m$ if $P(x) = \text{true}$

Security

$\mathcal{A} \xleftarrow{\text{mpk}} \mathcal{C}$ (mpk, msk)

$\xrightarrow{P}$ / $\xleftarrow{sk_P}$ ↻

$\xrightarrow{x^*, (m_0, m_1)}$ b: $\mathrm{Enc}(x^*, m_b) = ct^*$

$\xleftarrow{ct}$

$\xrightarrow{P}$ / $\xleftarrow{sk_P}$ ↻

For all efficient (poly. time) $\mathcal{A}$, $\left|\mathbb{P}[\mathcal{A}(\ldots) = b] - 1/2\right| \leq \text{negl}(\lambda)$ ($2^{-\lambda}$).

if for all P, $P(x) = \text{false}$.

Construction from LWE.

$\mathbb{Z}_q^{n \times m}$

FHE(GSW)/FH.Sig: $[A_i + x_i G] \xrightarrow{f} [A_f + f(x)G]$ — Hom. com. (/ciphertext)

FHE: $A_i := \begin{bmatrix} B \\ s^T B + e \end{bmatrix} \cdot R$ — randomizer.

FH. Sig. A_i with trapdoor.

C_1, C_2: $+$ $\quad C_1 + C_2 \quad (/\!/ A_+ = A_1 + A_2)$ ✓

$\times$ $\quad C_1 \cdot G^{-1}(C_2) \quad (/\!/ A_\times = A_1 \cdot G^{-1}(C_2))$

ABE: "Encrypt m under A_f" // ct

"Trapdoor for A_f" // sk.

$C_1 = A_1 + x_1 G$

$C_2 = A_2 + x_2 G$

$-A_1 \cdot G^{-1}(A_2) \quad x_1 A_2 + x_1 x_2 G$

$\boxed{C_\times = -C_1 \cdot G^{-1}(A_2) + x_1 C_2}$

$-A\, G^{-1}(A_2) - x_1 A$

$C_x = -C_1 \cdot G^{-1}(A_2) + x_1 C_2$

$-A_1 G^{-1}(A_2) - x_1 A_2$

$C_x = -A_1 G^{-1}(A_2) + x_1 x_2 G$

① Encode the policy checking
② How to encode the message.
③ Security?

$P(x) = \text{true}$
$= 0$

$s^T A_p + e \to s$

"Secret key": Trapdoor for A_p.

$A_i \longrightarrow A_p$

$s^T (A_i + x_i G) \longrightarrow s^T (A_p + P(x) G)$
(+noise) (+noise)

↳ recover s

$ct(x)$: $s^T (A_i + x_i G)$ (+noise)

$Dec(ct, sk_p)$: $s^T (A_p + P(x) G) \longrightarrow s$.

How to generate those trapdoors?

Trapdoor delegation.

Trapdoor for A: short matrix R
s.t. $A \cdot R = G$.

$s^T G + e$

$\begin{bmatrix} 1 & 2 \ldots & 2^{\lceil \log q \rceil - 1} \\ & \ddots & \\ & & \end{bmatrix}$

Claim 1: Given $s^T G + e$, it is easy to find s.
(Exercise: $q = 2^k$)

Claim 2: Can convert $s^T A + e \longrightarrow s^T G + e'$

Instead of trapdoor for A_p, $\bar{A}$
trapdoor for $[\bar{A} \| A_p]$.

Claim: Given a trapdoor for $\bar{A}$, can derive
trapdoor for $[\bar{A} \| A_p]$. // R s.t. $[\bar{A} \| A_p] R = G$

short.

trapdoor for $[\bar{A} \| A_p]$. // R s.t. $[\bar{A}\|A_p]R = G$.

$\neq \begin{bmatrix} R \\ 0 \end{bmatrix}$.

[~Micciancio-Peikert trapdoor]

Building $\bar{A}$

$\bar{A} = [A \| AR + G]$

$\bar{A} \cdot \begin{bmatrix} -R \\ I \end{bmatrix} = G$.

u: Given any u, can find short p s.t. $\bar{A}p = u$.

$p = \begin{bmatrix} -R \\ I \end{bmatrix} G^{-1}(u)$. //

$\bar{A} \cdot p = G \cdot G^{-1}(u) = u$.

$[\bar{A} \| A_p] \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = \underbrace{\bar{A} \cdot R_1}_{G - A_p R_2} + A_p R_2 = G$.

$R_2 \leftarrow$ short random

R_1 st $\bar{A} \cdot R_1 = G - A_p R_2$.

sample using trapdoor for $\bar{A}$

Scheme: Summary $\qquad$ h: length of attribute x.

$(P(x) = 0 \Rightarrow$ authorized$)$

Setup: $\left.\begin{array}{l} \bar{A}, \text{ with trapdoor} \\ \{A_i\}_{i \in h} \end{array}\right) \mathbb{Z}_q^{n \times m}$

mpk $= (\bar{A}, A_i)$

msk : trapdoor for $\bar{A}$

KeyGen(msk, P) : Derive A_p

// Derive trapdoor for $[\bar{A} \| A_p]$

Enc(mpk, x, m) : $s^T(\bar{A})$ + noise

$s^T(A_i + x_i G)$ + noise

Symmetric. key encrypt (s, m).

(Hard. core bits / LWE symmetric)

Dec(ct, sk_p) :

* Evaluate $\xrightarrow{x_i}$ $s^T(A_p + P(x)G)$ (+more noise)

use trapdoor to recover s

① If R is a trapdoor for A $\qquad$ correctness

$s^T A + e$

$R \rightarrow s$.

② Given trapdoor for $\bar{A}$ $\qquad$ correctness

"can derive trapdoor for $[\bar{A} \| A']$"

"looks independent of trapdoor for $\bar{A}$") security

A instantiation:

$\bar{A} = [A \| AR + G]$; $\bar{A} \begin{bmatrix} -R \\ I \end{bmatrix} = G$.

Delegation: $[\bar{A} \| A']$:

sample R_2, output $R_1 = \begin{bmatrix} -R \\ I \end{bmatrix} G^{-1}(G - A' R_2)$

output $[R_1 \ldots$

use trapdoor to recover s   (+more noise)

recover m.

sample R_2, output $R_1 = \begin{bmatrix} -R \\ I \end{bmatrix} G^{-1}(G - AR_2)$

output $\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$

Security. Way to generate trapdoors for all A_p s.t. $P(x) = 1$ (false).

$A_i \longrightarrow A_p$

$s^T(A_i + x_i G) + \text{noise} \longrightarrow s^T(A_p + P(x)G) + \text{more noise}$

$AR_i \longrightarrow AR_f$ $\|$ if $A_i = AR_i + x_i G$

$[A \,\|\, AR_p + \underset{\underset{P(x)=1}{\wedge}}{G}] \leftarrow R$ trapdoor.

Summary: Attribute-Based Encryption from LWE.

① "leveled" circuits. (bound on circuit depth) d

How to remove? Bootstrapping.

② ciphertexts $\text{poly}(\underbrace{k}_{|x|}, d)$

secret key $\text{poly}(d)$.

③ Weaker form of selective security.

Remove restriction from similar assumptions.