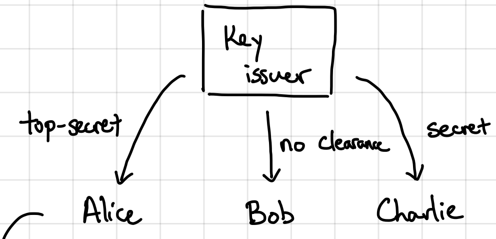
Attribute-based encryption (ABE): allow fine-grained access control to encrypted data



sk_{Alice} can be used to decrypt all messages that are "top secret", "secret", or "unclassified"

sk_{Bob} can be used to decrypt messages that are "unclassified" (but not "top secret" or "secret" messages)

Ciphertexts are associated with <u>attributes</u> x and a message μ

- $\text{Encrypt}(mpk, x, \mu) \rightarrow ct_{x,\mu}$

<u>public</u> attribute (e.g., "top-secret" or "secret" or "unclassified")

message

$$\text{Decrypt}(sk_f, ct_{x,\mu}) = \begin{cases} \mu & \text{if } f(x)=1 \\ \perp & \text{otherwise} \end{cases}$$

decryption succeeds if ciphertext attributes satisfy the decryption policy associated with the decryption key

More generally: keys are associated with <u>functions</u> (i.e, access control policies)

- $\text{KeyGen}(msk, f) \rightarrow sk_f$

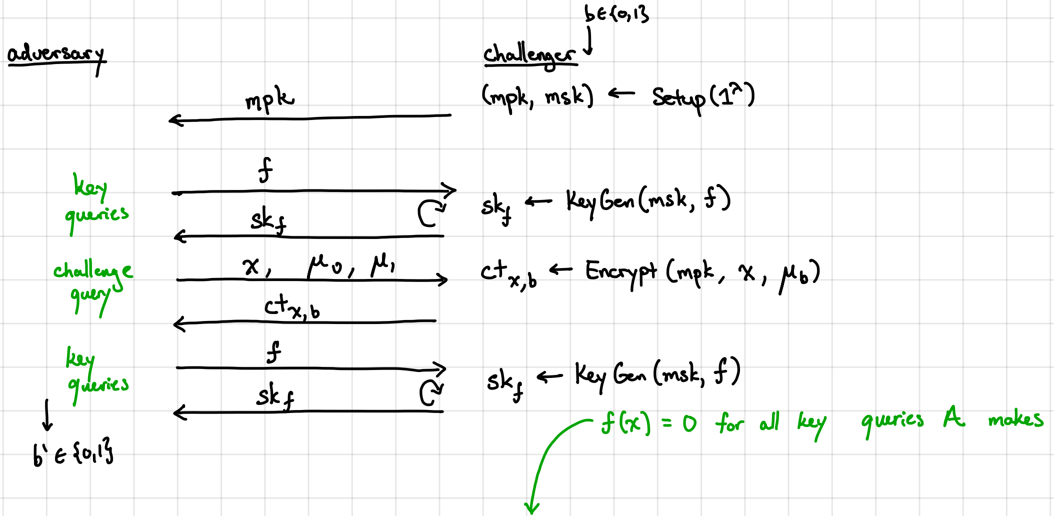
ABE Schema:

- $\text{Setup}(1^\lambda) \rightarrow mpk, msk$
- $\text{KeyGen}(msk, f) \rightarrow sk_f$
- $\text{Encrypt}(mpk, x, \mu) \rightarrow ct_{x,\mu}$
- $\text{Decrypt}(sk_f, ct_{x,\mu}) \rightarrow \mu$ or $\perp$

Correctness: for all functions f, attributes x where $f(x) = 1$, and all messages μ:

$$\Pr\left[\text{Decrypt}(sk_f, ct_{x,\mu}) = \mu \;\middle|\; \begin{array}{l} (mpk, msk) \leftarrow \text{Setup}(1^\lambda) \\ sk_f \leftarrow \text{KeyGen}(msk, f) \\ ct_{x,\mu} \leftarrow \text{Encrypt}(mpk, x, \mu \end{array}\right] = 1$$

Semantic Security:



$f(x) = 0$ for all key queries $\mathcal{A}$ makes

An ABE scheme is semantically secure if for all efficient and admissible adversaries $\mathcal{A}$,

$$|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]| \leq \text{negl}(\lambda)$$

__Starting point__: dual Regev encryption

KeyGen (1^λ): $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$

$r \xleftarrow{R} \{0,1\}^m$

$t \leftarrow Ar \in \mathbb{Z}_q^n$

pk: (A, t) sk: r

Encrypt (pk, μ): Sample $s \xleftarrow{R} \mathbb{Z}_q^n$, $e \leftarrow \chi^m$, $e' \leftarrow \chi$

Output $ct = (s^T A + e^T, s^T t + e' + \mu \cdot \lfloor \frac{q}{2} \rceil)$

Decrypt $(\overset{=r}{sk}, ct)$: Output $[ct_1 - ct_0 r]_2$

__Correctness__: $ct_1 - ct_0 r = s^T t + e' + \mu \cdot \lfloor \frac{q}{2} \rceil - s^T A r - e^T r$

$= \mu \cdot \lfloor \frac{q}{2} \rceil + \underbrace{e' - e^T r}_{\text{small}}$ ← if χ is B-bounded, then $|e' - e^T r| \leq B(m+1)$

correct as long as $B(m+1) \leq \frac{q}{4}$

__Security__: Follows from LHL and LWE: Hyb$_0$: real semantic security game

Hyb$_1$: sample $t \xleftarrow{R} \mathbb{Z}_q^n$ in the master public key

Hyb$_2$: sample $ct_0 \xleftarrow{R} \mathbb{Z}_q^m$, $ct_1 \xleftarrow{R} \mathbb{Z}_q$

↓ LHL (when $m = \Omega(n \log q)$)

↓ LWE

Comparison of primal vs. dual Regev:

primal Regev

pk: A, $b^T \leftarrow s^T A + e^T$

ct: Ar, $b^T r + \mu \cdot \lfloor \frac{q}{2} \rceil$

"interchanging" pk and ct

dual Regev

pk: A, $b \leftarrow Ar$

ct: $s^T A + e^T$

$s^T b + e' + \mu \cdot \lfloor \frac{q}{2} \rceil$

secret key is a __short__ preimage of public target vector b with respect to A

↳ will refer to this as dual Regev with respect to A

Attribute-based encryption from LWE: will "flip" the convention (decrypt when $f(x) = 0$, not when $f(x) = 1$).

__Idea__: suppose $x \in \{0,1\}^\ell$

public key will contain matrices $A \in \mathbb{Z}_q^{n \times m}$, $B = [B_1 | \cdots | B_\ell] \in \mathbb{Z}_q^{n \times \ell m}$

to encode an attribute $x \in \{0,1\}^\ell$:

$B - x \otimes G = [B_1 - x_1 G | \cdots | B_\ell - x_\ell G]$

then, to evaluate f on encodings:

$[B_1 - x_1 G | \cdots | B_\ell - x_\ell \cdot G] \cdot H_{f,x} = B_f - f(x) \cdot G$

only depends on function f (and $B_1, \ldots, B_\ell$) (independent of $\underline{x}$ — useful for key-generation)

when $f(x) = 0$ (can decrypt), we can recover B_f from $[B_1 - x_1 G | \cdots | B_\ell - x_\ell \cdot G]$

ciphertext will be a dual Regev ciphertext with respect to $[A | B_f]$:

mpk includes random vector $u \in \mathbb{Z}_q^n$

ciphertext is $s^T A + e^T$

$s^T [B_1 - x_1 G | \cdots | B_\ell - x_\ell \cdot G] + \tilde{e}^T$

$s^T u + e' + \mu \cdot \lfloor \frac{q}{2} \rceil$

will need to be careful with this distribution in security proof

$\xrightarrow{H_{f,x}}$ $s^T (B_f - f(x) \cdot G) + \tilde{e}^T H_{f,x}$

$= s^T B_f + \tilde{e}^T H_{f,x}$ when $f(x) = 0$

↳ decrypter can compute $s^T [A | B_f]$ + error

multiply by z_f yields $s^T u$ + error

secret key to a function f will be

short vector z_f such that $[A | B_f] z_f = u$

(can be sampled using trapdoor for A)

$[A | B_f]$ only depends on f and __not__ on input x

↳ secret key for a function f is a "recoding key": translates an LWE instance with respect to $[A | B_f]$ to LWE instance with respect to t: $[A | B_f] \cdot z_f = u$

Setup(1^λ): Define lattice parameters $n = n(\lambda)$, $q = q(\lambda)$, $m = \Theta(n \log q)$, $\chi = \chi(\lambda)$, $\sigma = \sigma(\lambda)$

$\uparrow$ error distribution

$\uparrow$ width parameter for preimage sampling (will set based on security proof – $s \sim m^{O(d)}$)

Sample $(A, T) \leftarrow \text{TrapGen}(n, q)$ $\quad A \in \mathbb{Z}_q^{n \times m}$

$B \xleftarrow{R} \mathbb{Z}_q^{n \times \ell m}$

$u \xleftarrow{R} \mathbb{Z}_q^{n}$

Output $mpk = (A, B, u)$

$msk = T$

KeyGen(mpk, msk, f): $B_f \leftarrow B \cdot H_f \in \mathbb{Z}_q^{n \times m}$ (input-independent evaluation)

$z_f \leftarrow [A \mid B_f]^{-1}(u)$

$\hookleftarrow \begin{bmatrix} T \\ 0 \end{bmatrix}$ is a trapdoor for $[A \mid B_f]$

output $sk_f \leftarrow z_f$

Encrypt(mpk, x, μ): Sample $s \xleftarrow{R} \mathbb{Z}_q^{n}$

Sample $e_1 \leftarrow \chi^m$, $e' \leftarrow \chi$, $R \xleftarrow{R} \{0,1\}^{m \times \ell m}$

Output $ct = \left(s^T A + e_1^T,\ s^T(B - x \otimes G) + e_1^T R,\ s^T u + e' + \mu \cdot \lfloor \frac{q}{2} \rceil,\ x\right)$

Decrypt$(\overset{= z_f}{sk_f}, \underset{(ct_1, ct_2, ct_3)}{\underset{\shortparallel}{ct}})$: compute $ct_3 - [ct_1 \mid ct_2 H_{f,x}] z_f$ and round

Correctness. Suppose $f(x) = 0$. Then

$$\left(s^T(B - x \otimes G) + e_1^T R\right) H_{f,x} = s^T\left(B_f - f(x) \cdot G\right) + e_1^T R H_{f,x}$$
$$= s^T B_f + e_1^T R H_{f,x} \quad \text{since } f(x) = 0$$

Next: $\left(s^T [A \mid B_f] + [e_1^T \mid e_1^T H_{f,x}]\right) z_f$

$$= s^T u + [e_1^T \mid e_1^T H_{f,x}] z_f$$

Thus, we compute

$$\mu \cdot \lfloor \tfrac{q}{2} \rceil + \underbrace{e' - [e_1^T \mid e_1^T H_{f,x}] z_f}$$

"small" since, e_1, e' are from noise distribution and $\|H_{f,x}\| \leq (n \log q)^{O(d)}$ where d is the depth of the computation

Security. Proving security is delicate. Need to be able to simulate decryption keys, but we do not have a trapdoor for A (otherwise LWE is easy).

- $\hookrightarrow$ In other words, if x is the challenge attribute, we need to be able to give out keys for all functions f where $f(x) = 1$ but be unable to give out keys for $f(x) = 0$.
- $\hookrightarrow$ Key technique: "punctured trapdoor" that works only for functions f where $f(x) = 1$.

To leverage this technique, we will consider selective security where adversary has to declare attribute before seeing public parameters

Open problem: Adaptively-secure ABE from polynomial hardness of LWE

Proof of Security. We will consider a sequence of experiments:

Hyb$_0$: real security game encrypting μ_0

Hyb$_1$: after adversary selects the challenge attribute $x^* \in \{0,1\}^\ell$, challenger constructs the public key as follows: $(A, T) \leftarrow \text{TrapGen}(n, q)$

$$R \xleftarrow{R} \{0,1\}^{m \times m\ell}$$
$$B = AR + (x^* \otimes G)$$

$\text{mpk} = (A, B, u)$ where $u \xleftarrow{R} \mathbb{Z}_q^n$

to answer key-generation queries for f, challenger computes

$$B_f \leftarrow B \cdot H_f$$
$$z_f \leftarrow [A \mid B_f]^{-1}(u) \text{ with trapdoor } \begin{bmatrix} T \\ 0 \end{bmatrix}$$

to construct the challenge ciphertext, challenger samples $s \xleftarrow{R} \mathbb{Z}_q^n$, $e_1 \leftarrow \chi^m$, $e' \leftarrow \chi$ and outputs $ct = (s^T A + e_1^T, s^T(B - x^* \otimes G) + e_1^T R, s^T u + e' + \mu \cdot \lfloor \frac{q}{2} \rceil, x^*)$

Hyb$_0$ and Hyb$_1$ are statistically indistinguishable by LHL [need a variant where $(A, AR, e^T R) \overset{s}{\approx} (A, u, e^T R)$]

$\leftarrow$ $e^T R$ is partial leakage on R

(statement holds for all e when $m > 2n \log q$)

Hyb$_2$: key-generation queries are answered without using trapdoor for A: instead, challenger computes $R_{f,x^*} = R \cdot H_{f,x^*}$ and outputs

$$z_f \leftarrow [A \mid B_f]^{-1}(u) \text{ using trapdoor } \begin{bmatrix} -R_{f,x^*} \\ I \end{bmatrix}$$

Observe: $(B - x^* \otimes G) H_{f,x^*} = B_f - f(x^*) \cdot G$

Adversary can only query on x^* where $f(x^*) = 1$ (policy is unsatisfied).

$$\Rightarrow (B - x^* \otimes G) H_{f,x^*} = B_f - G$$
$$\parallel$$
$$AR + (x^* \otimes G) - (x^* \otimes G) = AR$$

$$\Rightarrow ARH_{f,x^*} = B_f - G \Rightarrow [A \mid B_f] \cdot \begin{bmatrix} -R_{f,x^*} \\ I \end{bmatrix} = G$$

critical here that $f(x^*) = 1$

otherwise, we end up with $[A \mid B_f] \cdot \begin{bmatrix} -R_{f,x^*} \\ I \end{bmatrix} = 0$

not a trapdoor since $\begin{bmatrix} -R_{f,x^*} \\ I \end{bmatrix}$ not full rank over the reals

Key observation: Trapdoor only works if $f(x^*) = 1$. If $f(x^*) = 0$, then $AR_{f,x^*} = B_f$ and we do not have a trapdoor for $[A \mid B_f]$. Referred to as a "punctured" trapdoor.

Hyb$_3$: replace challenge ciphertext with $(z_1^T, z_1^T R, z', x^*)$ where $z_1 \xleftarrow{R} \mathbb{Z}_q^m$, $z' \xleftarrow{R} \mathbb{Z}_q$

follows by LWE since challenge ciphertext is now

$$s^T A + e_1^T$$
$$s^T (B - (x^* \otimes G)) + e_1^T R = s^T AR + e_1^T R = (s^T A + e_1^T) R$$
$$s^T u + e' + \mu_0 \cdot \lfloor \tfrac{q}{2} \rceil$$

apply LWE to $s^T A + e_1^T$ and $s^T u + e'$