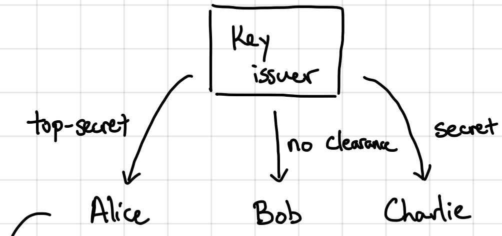
Attribute-based encryption (ABE): allow fine-grained access control to encrypted data



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 attributes x and a message μ

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

public 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 functions (i.e, access control policies)

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

ABE Schema:

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

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:

adversary | challenger ($b \in \{0,1\}$)

- challenger: $(mpk, msk) \leftarrow \text{Setup}(1^\lambda)$; challenger → adversary: mpk
- key queries: adversary → challenger: f; challenger: $sk_f \leftarrow \text{KeyGen}(msk, f)$; challenger → adversary: sk_f
- challenge query: adversary → challenger: x, μ_0, μ_1; challenger: $ct_{x,b} \leftarrow \text{Encrypt}(mpk, x, \mu_b)$; challenger → adversary: $ct_{x,b}$
- key queries: adversary → challenger: f; challenger: $sk_f \leftarrow \text{KeyGen}(msk, f)$; challenger → adversary: sk_f
- adversary outputs $b' \in \{0,1\}$

$f(x) = 0$ 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 $ct = (s^TA + e^T, s^Tt + e' + \mu \cdot \lfloor \frac{q}{2} \rceil)$

Decrypt$(\overset{=r}{sk}, ct)$: Output $\lfloor ct_1 - ct_0 r \rceil_2$

Correctness: $ct_1 - ct_0 r = s^Tt + e' + \mu \cdot \lfloor \frac{q}{2} \rceil - s^TAr - e^Tr$

$= \mu \cdot \lfloor \frac{q}{2} \rceil + \underbrace{e' - e^Tr}_{\text{small}}$

if χ is B-bounded, then

$|e' - e^Tr| \leq B(m+1)$

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:

"interchanging" pk and ct

primal Regev	dual Regev
pk: A, $b^T \leftarrow s^TA + e^T$	pk: A, $b \leftarrow Ar$
ct: Ar, $b^Tr + \mu \cdot \lfloor \frac{q}{2} \rceil$	ct: $s^TA + e^T$ $s^Tb + e' + \mu \cdot \lfloor \frac{q}{2} \rceil$

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 $A, B_1, \ldots, B_\ell \in \mathbb{Z}_q^{n\times m}$

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

$[B_1 - x_1 G \mid \cdots \mid B_\ell - x_\ell G]$

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

then, to evaluate f on encodings:

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

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

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

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

ciphertext is $s^TA + e^T$

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

$s^Tt + e' + \mu \cdot \lfloor \frac{q}{2} \rceil$

will need to be careful with this distribution in security proof

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

$= s^TB_f + \tilde{e}^TH_{f,x}$ when $f(x) = 0$

Secret key to a function f will be

short vector z_f such that $[A \mid B_f] z_f = t$

(can be sampled using trapdoor for A)

↳ decrypter can compute

$s^T[A \mid B_f]$ + error

multiply by z_f yields

s^Tt + error

$[A \mid B_f]$ only depends on f and not on input x