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

- **Key issuer**
  - Alice
  - Bob
  - Charlie

Ciphertexts are associated with attributes $x$ and a message $\mu$:
- $\text{Encrypt}(\text{mpk}, x, \mu) \rightarrow \text{ct}_{x,\mu}$

More generally: keys are associated with functions (i.e., access control policies):
- $\text{KeyGen}(\text{msk}, f) \rightarrow \text{sk}_f$

**ABE Scheme:**
- $\text{Setup}(\lambda^2) \rightarrow \text{mpk, msk}$
- $\text{KeyGen}(\text{msk}, f) \rightarrow \text{sk}_f$
- $\text{Encrypt}(\text{mpk}, x, \mu) \rightarrow \text{ct}_{x,\mu}$
- $\text{Decrypt}(\text{sk}_f, \text{ct}_{x,\mu}) \rightarrow \mu$ or 1

**Correctness:** for all functions $f$, attributes $x$ where $f(x) = 1$, and all messages $\mu$:
$$\Pr\left[\text{Decrypt}(\text{sk}_f, \text{ct}_{x,\mu}) = \mu \mid (\text{mpk, msk}) \leftarrow \text{Setup}(\lambda^2), \text{sk}_f \leftarrow \text{KeyGen}(\text{msk}, f), \text{ct}_{x,\mu} \leftarrow \text{Encrypt}(\text{mpk}, x, \mu)\right] = 1$$

**Semantic Security:**

An ABE scheme is semantically secure if for all efficient and admissible adversaries $A$, 
$$\left|\Pr[\hat{b}' = 1 \mid b = 0] - \Pr[\hat{b}' = 1 \mid b = 1]\right| \leq \negl(\lambda)$$
Starting point: dual Regev encryption

Key Gen (\(2^n\)): \(A \in \mathbb{Z}_0^n\), \(r \in \{0,1\}^n\)
\(t \leftarrow Ar \in \mathbb{Z}_0^n\)
\(pk: (A, t) \quad sk: r\)

Encrypt \((pk, m)\): Sample \(s \in \mathbb{Z}_0^n\), \(e \leftarrow \mathbb{X}^n\), \(e' \leftarrow \mathbb{X}\)
Output \(ct = (s^T A + e^T, s^T t + e' + \mu \cdot \left| \frac{s}{2} \right|)\)

Decrypt \((sk, ct)\): Output \([ct_1 - ct_2 \cdot r]\)

Correctness: \(ct_1 - ct_2 \cdot r = s^T t + e' + \mu \cdot \left| \frac{s}{2} \right| - s^T Ar - e'r\)
\(= \mu \cdot \left| \frac{s}{2} \right| + e' - e'r\) if \(X\) is \(B\)-bounded, then \(\left| e' - e'r \right| \leq B(m+1)\), correct as long as \(B(m+1) \leq \frac{\mu}{2}\)

Security: Follows from LHL and LWE:
Hyb\(_a\): real semantic security game
Hyb\(_b\): sample \(t \in \mathbb{Z}_0^n\) in the master public key
Hyb\(_b\): sample \(ct_0 = s^T t, ct_1 \in \mathbb{Z}_0^n\)

Comparison of primal vs. dual Regev:

primal Regev
\(pk: A, b^t = s^T A + e^T\)
\(ct: Ar, b^T t + \mu \cdot \left| \frac{s}{2} \right|\)

"interchanging" ph and \(ct\)

dual Regev
\(pk: A, b \leftarrow Ar\)
\(ct: s^T A + e^T\)
\(s^T b + e' + \mu \cdot \left| \frac{s}{2} \right|\)

secret key is a short package of public target vector \(b\) with respect to \(A\)
\(\rightarrow\) 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 \(x \in \{0,1\}^d\)
public key will contain matrices \(A \in \mathbb{Z}_0^m\), \(B = \begin{bmatrix} B_1 & \cdots & B_n \end{bmatrix} \in \mathbb{Z}_0^{m \times n}\)

to encode an attribute \(x \in \{0,1\}^d\):
\(B - x \otimes G = [B_1 - x_1 G \mid \cdots \mid B_n - x_n G]\)

then, to evaluate \(f\) on encodings:
\([B_1 - x_1 G \mid \cdots \mid B_n - x_n G]: H_{f,x} = B_y - f(x): G\)

when \(f(x) = 0\) (can decrypt), we can recover \(B_y\) from \([B_1 - x_1 G \mid \cdots \mid B_n - x_n G]\)
ciphertext will be a dual Regev ciphertext with respect to \([A, B_y]\):

\(\left[A, B_y\right]\) only includes random vector \(u \in \mathbb{Z}_0^n\)
ciphertext is \(s^T A + e^T\)
\(s^T \left[B_1 - x_1 G \mid \cdots \mid B_n - x_n G\right] + e^T\)
\(s^T u + e' + \mu \cdot \left| \frac{s}{2} \right|\)

secret key to a function \(f\) will be short vector \(z\) such that \([A, B_y]z = u\)
(can be sampled using trapdoor for \(A\))

\(H_{f,x} = s^T \left[B_1 - f(x):G\right] + e^T H_{f,x}\)
\(= s^T B_y + e^T H_{f,x}\) when \(f(x) = 0\)

\(\rightarrow\) decryptor can compute \(s^T [A, B_y] + \text{error}\)
multiply by \(z\) yields \(s^T u + \text{error}\)

\([A, B_y]\) only depends on \(f\) and not on input \(x\)

\(\uparrow\) secret key for a function \(f\) is a "reading key" translates an LWE instance with respect to \([A, B_y]\) to LWE instance with respect to \(A\)
\(t = [A, B_y]: z = u\)
Setup $(\Lambda)$: Define lattice parameters $n = n(x)$, $q = q(x)$, $m = O(n \log q)$, $\chi \times x$, $\sigma = \sigma(x)$

Sample $(A, T) \leftarrow \text{TrapGen}(n, q)$
$B \leftarrow Z_{q^n}^{*}$
$u \leftarrow Z_q^*$
Output $\text{mpk} = (A, B, u)$

\text{msk} = T

\text{KeyGen}(\text{mpk}, \text{msk}, f)$:
$B_f \leftarrow B \cdot H_f \in Z_{q^n}$ (input-independent evaluation)
\[
\tilde{z}_f \leftarrow [A \mid B_f]^{*}(u)
\]\[
\tilde{z}_f \text{'s a trapdoor for } [A \mid B_f]
\]
Output $\text{sky} = \tilde{z}_f$

\text{Encrypt}(\text{mpk}, X, \mu)$:
Sample $s \leftarrow Z_q^{*}$
Sample $e_1 \leftarrow X$, $e \leftarrow X$, $R \leftarrow Z_{q^n}$
Output $c^* = (s^T A \cdot e_1^T, s^T (B - x \otimes G) + e^T R, s^T u + e' + \mu \cdot [\frac{9}{12}], x)$

\text{Decrypt}(\text{sky}, c^*)$:
Compute $c^* = [c_t | c_{t +} H_f, x] z_f$ and round $\text{ct}_t = [c_t, c_t H_f, x] z_f$ and round

Correctness. Suppose $f(x) = 0$. Then
\[
(s^T (B - x \otimes G) + e^T R) H_f, x = s^T (B_f - f(x) \cdot G) + e^T R H_f, x
\]
\[
= s^T B_f + e^T R H_f, x \text{ since } f(x) = 0
\]
Next:
\[
(s^T [A \mid B_f] + [e_1^T | e_1^T H_f, x]) z_f
\]
\[
= s^T u + [e_1^T | e_1^T H_f, x] z_f
\]
Thus, we compute
\[
\mu \cdot [\frac{9}{12}] + e' = [e_1^T | e_1^T H_f, x] z_f
\]
"small" since, $e_1, e'$ are from noise distribution and
\[
||H_f|| \leq \Theta(n \log q)^{O(1)} \text{ where } d \text{ is the depth of the computation}
\]

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).

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 $f(x) = 0$.

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:

Hyb1: real security game encrypting \( \mu \)

Hyb2: after adversary selects the challenge attribute \( x^* \in \{0,1\}^S \), challenger constructs the public key as follows: \((A, T) \leftarrow \text{TrapGen}(n, g) \)
\[ R \leftarrow \{0,1\}^{|T|} \]
\[ B = AR + (x^* \otimes G) \]

\[ \text{mpk} = (A, B, u) \text{ when } U \leftarrow \{0,1\}^B \]

to answer key-generation queries for \( f \), challenger computes
\[ B_f \leftarrow B \cdot H_f \]
\[ z_f \leftarrow [A \mid B_f]^t(u) \text{ with trapdoor } [B] \]

to construct the challenge ciphertext, challenger samples \( s \leftarrow \{0,1\}^S, e_1 \leftarrow \mathbb{R}, e \leftarrow \mathbb{R} \)

and outputs \( C_1 = (s^T A + e_1, s^T(B - x^* \otimes G) + e R, sT u + e + \mu . [3], x^*) \)

Hyb1 and Hyb2 are statistically indistinguishable by LHL

Hyb2: key-generation queries are answered without using trapdoor for \( A \):

instead, challenger computes \( R_{z_{*,x}} = R \cdot H_f x \)

and outputs \( z_f \leftarrow [A \mid B_f]^{-1}(u) \) using trapdoor \([R] \)

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

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

If \( f(x^*) = 2 \), then \( f(x^*) \neq 1 \) and we do not have a trapdoor for \([A \mid B_f]\). Referred to as a "punctured" trapdoor.

\[ B_f \text{ replace challenge ciphertext with } (z^T, z^T R, z', x^*) \text{ where } z \leftarrow \mathbb{R}^m, z' \leftarrow \mathbb{R}^n \]

follow by LWE since challenge ciphertext is now
\[ s^T A + e_1 \]
\[ s^T(B - (x^* \otimes G)) + e R = s^T AR + e R - (s^T A + e_1) R \]
\[ s^T u + e' + \mu s . [3] \]

apply LWE to \( s^T A + e_1 \) and \( s^T u + e' \).