Decouple into two equations:

- Input-independent evaluation: $B_1, \ldots, B_\ell, f \mapsto B_f$
- Input-dependent evaluation: $[B_1 - x_1 G \mid \cdots \mid B_\ell - x_\ell G] H_{f,x} = B_f - f(x) \cdot G$

] Will give us many advanced primitives!

<u>Unforgeability</u>: Will consider a weaker (selective) notion of security where the message that is signed is <u>independent</u> of the verification key [not difficult to get full adaptive security, but somewhat tedious]

<u>adversary</u> | <u>challenger</u>

adversary $\xrightarrow{x}$ challenger

$(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$

$\sigma_x \leftarrow \text{Sign}(sk, x)$

adversary $\xleftarrow{vk}$ challenger

adversary $\xleftarrow{\sigma_x}$ challenger

adversary $\xrightarrow{f,\, y,\, \sigma_{f,y}}$ challenger

$\downarrow$

Output 1 if $y \neq f(x)$ and $vk_f \leftarrow \text{Preprocess}(vk, f)$, $\text{Verify}(vk_f, y, \sigma_{f,y}) = 1$

<u>Proof of unforgeability.</u> (from LWE)

Can also give a proof from SIS. [HW Exercise]

Selective security: message is programmed into the vk

Hyb_0: real signature unforgeability game

Hyb_1: instead of sampling $B_1, \ldots, B_\ell$ uniformly, sample $R_1, \ldots, R_\ell \leftarrow \{0,1\}^{m \times t}$ and set $B_i \leftarrow AR_i + x_i G$

it sets $vk = (A, B_1, \ldots, B_\ell)$ and $\sigma = (R_1, \ldots, R_\ell)$

Hyb_2: instead of sampling (A, T) using TrapGen, challenger samples

$$\bar{A} \xleftarrow{R} \mathbb{Z}_q^{(n-1) \times m}, \quad s \xleftarrow{R} \mathbb{Z}_q^{n-1}, \quad e \leftarrow \chi^m \quad \rightsquigarrow \quad A \leftarrow \begin{bmatrix} \bar{A} \\ s^T \bar{A} + e \end{bmatrix}$$

(LWE matrix)

$\text{Hyb}_0 \overset{s}{\approx} \text{Hyb}_1$ by LHL

$\text{Hyb}_1 \overset{c}{\approx} \text{Hyb}_2$ under LWE

Suppose $\mathcal{A}$ succeeds in Hyb_2. Namely $\mathcal{A}$ outputs R^*, f, y such that

$$AR^* = B_f - y \cdot G$$

where $y \neq f(x)$ and R^* is short. Let

$$R_{f,x} = [R_1 \mid \cdots \mid R_\ell] H_{f,x}$$

Then, by construction,

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

Then, $AR^* - AR_{f,x} = (-1)^c \cdot G$ where $c \in \{0,1\}$ and $\underbrace{[-s^T \mid 1](AR^* - AR_{f,x})}_{\underbrace{e^T (R^* - R_{f,x})}_{\text{small relative to } q \text{ since } e,\, R^*,\, R_{f,x} \text{ small}}} = [-s^T \mid 1] \cdot G$

$$= [-s^T \mid 1] \cdot G$$

has large entries $(\frac{q}{2})$

Implication: in Hyb_2, scheme is <u>statistically</u> unforgeable!

<u>Proof technique</u>: programmed x into the public parameters

<u>statistically</u> unforgeable at $f(x)$ for all f

↳ could be forgeable on other messages: "somewhere unforgeable"

Context-hiding for homomorphic signatures:

- In many settings, we also want the computed signature to hide information about the input to the computation

Alice $\xrightarrow{x,\ \sigma_x}$ Server

Server $\xleftarrow{f}$ Bob

Server $\xrightarrow{y = f(x),\ \sigma_{f,y}}$ Bob

Bob wants to check signature on $y = f(x)$ but should not learn anything about x

- We will see one application of this type of property to (designated-prover) NIZKs

We say a homomorphic signature scheme is statistically context-hiding if there exists an efficient simulator S where for all $(vk, sk) \leftarrow \mathrm{KeyGen}(1^\lambda)$, $x \in \{0,1\}^\ell$, and $f: \{0,1\}^\ell \to \{0,1\}$:

$$\{vk, \mathrm{Eval}(vk, f, \sigma)\} \overset{s}{\approx} \{vk, S(sk, vk, f, f(x))\}$$

simulator needs to simulate valid signatures so it needs to know the signing key; however, it does not know the input x, only the value $f(x)$

this means signature reveals no information about x other than $(f, f(x))$.

Turns out this is not difficult to achieve!

Current construction is not context-hiding:

$$R_{f,x} := [R_1 | \cdots | R_\ell] \cdot H_{f,x}$$

this is a function of x!

To achieve context-hiding, we need a way to re-randomize a signature.

Suppose $AR_{f,x} = B_f - y \cdot G$ where $y \in \{0,1\}$

Evaluator knows y so it can compute the matrix

$$V := [A \mid B_f + (y-1) \cdot G] = [A \mid AR_{f,x} + (2y-1) \cdot G]$$

Now, since $y \in \{0,1\}$, $2y - 1 \in \{-1,1\}$. Then $R_{f,x}$ is a trapdoor for V:

$$V \cdot \begin{bmatrix} -R_{f,x} \\ I \end{bmatrix} = (2y-1) \cdot G = G \text{ or } -G$$

The public key then includes a random target $z \xleftarrow{R} \mathbb{Z}_q^n$ and the signature is formed by sampling a short vector t such that $Vt = z$:

$$t \leftarrow \mathrm{SamplePre}(V, \pm R_{f,x}, z, s) \quad \text{for some } s = (n \log q)^{O(d)}$$

To verify a signature, the verifier computes B_f from $B_1, \ldots, B_\ell$, constructs V from the verification key and checks that

$$Vt = z \quad \text{and} \quad \|t\|_\infty \leq \beta \quad \text{where } \beta = (n \log q)^{O(d)} \text{ is the noise bound}$$

For context-hiding, we observe that $t \sim D_{\mathcal{L}_z^\perp(H), s}$ where H only depends on $A, B_1, \ldots, B_\ell, f, y$ (independent of x)

$\hookrightarrow$ We can sample from $D_{\mathcal{L}_z^\perp(H), s}$ using a trapdoor for A (since V is an extension of A — see HW 1)

__Unforgeability__: Follows by similar argument as before.

We argue selective security from LWE as before (can also argue security from SIS — see HW1)

Hyb_0: real unforgeability game

Hyb_1: after adversary chooses $x \in \{0,1\}^\ell$,

sample verification key as $[B_1 | \cdots | B_\ell] = A[R_1 + x_1 G | \cdots | R_\ell + x_\ell \cdot G]$

output $vk = (A, B_1, \ldots, B_\ell, z)$ where $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $z \xleftarrow{R} \mathbb{Z}_q^n$

Hyb_2: Sample A as an LWE matrix

$$A \leftarrow \begin{bmatrix} \bar{A} \\ \bar{s}^T \bar{A} + e^T \end{bmatrix} \quad \text{where } \bar{A} \xleftarrow{R} \mathbb{Z}_q^{(n-1)\times m} \quad e \xleftarrow{R} \chi^m \quad \bar{s} \xleftarrow{R} \mathbb{Z}_q^{n-1}$$

We show in Hyb_2 that advantage of __any__ adversary $\mathcal{A}$ is negligible:

Suppose $\mathcal{A}$ outputs signature $\sigma = t$ on (f, y) where $y \neq f(x)$. $[f(x) = 1-y]$

Let $V = [A | B_f + (y-1) \cdot G]$. Then $Vt = z$ (since signature verifies).

Let $R_{f,x} = [R_1 | \cdots | R_\ell] \cdot H_{f,x}$. Then

$$AR_{f,x} = A[R_1 | \cdots | R_\ell] \cdot H_{f,x} = [B_1 - x_1 G | \cdots | B_\ell - x_\ell G] H_{f,x}$$
$$= B_f - f(x) \cdot G$$
$$= B_f - (1-y) \cdot G$$

$$\Longrightarrow B_f = AR_{f,x} + (1-y) \cdot G$$

Then $V = [A | B_f + (y-1) \cdot G] = [A | AR_{f,x}]$ so $Vt = [A | AR_{f,x}] t$.

If $Vt = z$, then

$$\underbrace{s^T [A | AR_{f,x}] t}_{\text{small (norm bounded by } (n \log q)^{O(d)})} = \boxed{s^T z} \qquad (s = [-\bar{s} | 1])$$

$$s^T [A | AR_{f,x}] t = \underbrace{[e^T | e^T R_{f,x}] t}_{\text{small (norm bounded by } (n \log q)^{O(d)})}$$

$s^T z$: large so long as $q \gg (n \log q)^{O(d)}$

__Recap__:

__homomorphic encryption__

$$pk: A = \begin{bmatrix} \bar{A} \\ s^T \bar{A} + e^T \end{bmatrix}$$

$$ct: \underbrace{C}_{\text{ciphertext}} = A\underbrace{R}_{\text{encryption randomness}} + \underbrace{\mu}_{\text{message}} \cdot G$$

__homomorphic signatures__

$$vk: A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$$

$$\text{signature}: A\underbrace{R}_{\text{signature}} = \underbrace{B}_{\text{target matrix (in vk)}} - \underbrace{\mu}_{\text{message}} G$$

GSW homomorphisms are homomorphic on both messages __and__ on randomness

$$C_1, \ldots, C_\ell, f \longmapsto C_f$$

$$[C_1 - x_1 G | \cdots | C_\ell - x_\ell \cdot G] \cdot H_{f,x} = C_f - f(x) \cdot G$$

$$\| $$

$$A[R_1 | \cdots | R_\ell] H_{f,x} \rightsquigarrow [R_1 | \cdots | R_\ell] H_{f,x} = R_{f,x}$$

$$C_f = AR_{f,x} + f(x) \cdot G$$

homomorphism on message

homomorphism on randomness

HE: ciphertext evaluation

HS: verification

HS: signature evaluation