Next up: homomorphic signatures

client | **server**

$\sigma \leftarrow \text{Sign}(vk, x)$

$\xrightarrow{x,\ \sigma}$

$\xrightarrow{f}$ $\quad y \leftarrow f(x)$

$\xleftarrow{y,\ \sigma_{f,y}}$ $\quad \sigma_y \leftarrow \text{Eval}(f, x, \sigma)$

↓

checks that $\sigma_{f,y}$ is a signature on y with respect to function f

↳ can view as signature on pair (f, y) ← Why not just on y alone?

Requirements: Unforgeability: Cannot construct signature σ on (f, y) where $y \neq f(x)$.
(Will formalize later)

Succinctness: Size of $\sigma_{f,y}$ should be $|y| \cdot \text{poly}(\lambda)$. In particular, should not depend on $|x|$ or $|f|$.
↳ Otherwise trivial to construct! (Outputting $(\sigma, x, f(x))$ suffices).

Efficient verification: Can decompose verification algorithm as follows: ↳ Also the case for FHE!

- $\text{Preprocess}(vk, f) \to vk_f$ — Generates short function verification key vk_f ($|vk_f| = \text{poly}(\lambda, d)$) — depth of circuit computing f
- $\text{Verify}(vk_f, y, \sigma) \to 0/1$ — Runs in time $\text{poly}(\lambda, d, |y|)$

Homomorphic signatures allow computations on authenticated data.

Defining unforgeability: adversary | challenger

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

$\xleftarrow{vk}$

$\xrightarrow{x}$

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

One-time security (generalizes to many-time)

$\xleftarrow{\sigma_x}$

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

↓

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

Construction: relies on similar homomorphic structure as GSW (for message space $\{0,1\}^\ell$)

- $\text{KeyGen}(1^\lambda)$: Set lattice parameters $n = n(\lambda)$, $q = q(\lambda)$.
  Sample $(A, T) \leftarrow \text{TrapGen}(n, q)$ $\quad [A \in \mathbb{Z}_q^{n \times m},\ T \in \{0,1\}^{m \times t}]$
  ↳ $AT = G \in \mathbb{Z}_q^{n \times t}$; $t = n \lceil \log q \rceil$
  Sample $B_1, \ldots, B_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times t}$
  Output $vk = (A, B_1, \ldots, B_\ell)$, $sk = R$
- $\text{Sign}(sk, x)$: Compute $R_i \leftarrow A^{-1}(B_i - x_i G)$ for $i \in [\ell]$ using T
  In particular:
  $$A[R_1 | \cdots | R_\ell] = [B_1 - x_1 G | \cdots | B_\ell - x_\ell G] \qquad (R_i \in \mathbb{Z}_q^{m \times t})$$
  $$= [B_1 | \cdots | B_\ell] - x \otimes G$$
  Output $\sigma = (R_1, \ldots, R_\ell)$
- $\text{Verify}(vk, x, \sigma)$: Check that $\|R_i\| \leq B$ and that $A[R_1 | \cdots | R_\ell] \stackrel{?}{=} [B_1 | \cdots | B_\ell] - x \otimes G$
  ↳ bound based on quality of trapdoor (lattice parameters)

Homomorphic evaluation: $\quad A[\underbrace{R_1 | \cdots | R_\ell}_{\text{signatures}}] = [\underbrace{B_1 - x_1 G | \cdots | B_\ell - x_\ell G}_{\text{verification keys}}]$

To derive a signature on the sum of two bits $(x_i + x_j)$:

$$\left.\begin{aligned} R_+ &= R_i + R_j \\ B_+ &= B_i + B_j \end{aligned}\right\} \text{Verification: } \underbrace{AR_+}_{\text{new signature}} \stackrel{?}{=} \underbrace{B_+}_{\text{new verification component associated with addition operation}} - (x_i + x_j)G$$

To derive a signature on the product of two bits $(x_i x_j)$:

$$\begin{aligned} AR_i &= B_i - x_i G \\ AR_j &= B_j - x_j G \end{aligned} \quad \Rightarrow \quad \text{desire something of the form} \quad AR_\times = B_\times - x_i x_j \cdot G$$

$AR_\times$: function of R_i, R_j and x_i, x_j (should be short)

$B_\times$: function of B_i, B_j — should not depend on x_i, x_j (verification algorithm does not know x)

$$AR_i = B_i - x_i G \longrightarrow B_i = AR_i + x_i G$$

$$\begin{aligned} AR_j G^{-1}(B_i) &= (B_j - x_j \cdot G)\, G^{-1}(B_i) \\ &= B_j G^{-1}(B_i) - x_j B_i \\ &= B_j G^{-1}(B_i) - A(x_j R_i) - x_i x_j G \end{aligned}$$

$$\Rightarrow A\underbrace{(R_j G^{-1}(B_i) + x_j R_i)}_{R_\times = R_j G^{-1}(B_i) + x_j R_i} = \underbrace{B_j G^{-1}(B_i)}_{B_\times = B_j G^{-1}(B_i)} - x_i x_j \cdot G$$

$R_\times$: function of signature, input; $\|R_\times\|_\infty \le \|R_j\|_\infty \cdot t + \|R_i\|_\infty$

$B_\times$: function of public key only (this is GSW homomorphic multiplication)

Observation:

$$\begin{aligned} R_+ &= R_i + R_j &&= [R_i \mid R_j] \begin{bmatrix} I_t \\ I_t \end{bmatrix} \\ R_\times &= R_i (x_j I_t) + R_j G^{-1}(R_i) &&= [R_i \mid R_j] \begin{bmatrix} x_j I_t \\ G^{-1}(R_i) \end{bmatrix} \end{aligned}$$

small linear function of R_i and R_j — can depend on R_i, R_j, x

Compose above operations to compute signature on $R_{f,x}$ on evaluation $f(x)$

By above analysis, multiplication scales noise by a factor of t so if f can be computed by a circuit of depth d, $\|R_{f,x}\|_\infty \le t^{O(d)}$

To verify a signature $R_{f,x}$ on $(f, z = f(x))$, verifier computes B_f from $B_1, \ldots, B_\ell$ and checks that $\|R_{f,x}\|_\infty$ sufficiently small (bound $\sim t^{O(d)}$)

$$AR_{f,x} = B_f - z \cdot G$$

this can also be written as $B_f \leftarrow [B_1 | \cdots | B_\ell] \cdot H_f$ where $\|H_f\| \le m^{O(d)}$ and depends only on $B_1, \ldots, B_\ell, f$

More generally:

$$R_{f,x} = [R_1 | \cdots | R_\ell] \cdot H_{f,x} \quad \text{where } H_{f,x} \in \mathbb{Z}_q^{\ell t \times t} \text{ and } \|R_{f,x}\|_\infty \le t^{O(d)} = (n \log q)^{O(d)}$$

where d is the (multiplicative) depth of the circuit computing f

Now, if $AR_i = B_i - x_i G$, then from the above,

$$AR_{f,x} = B_f - f(x) \cdot G$$

where B_f is the matrix obtained by evaluating f on $\overset{\approx AR_1}{B_1}, \ldots, \overset{\approx AR_\ell}{B_\ell}$

This can be expanded as

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

Decouple into two equations:

- Input-independent evaluation: $[B_1 \mid \cdots \mid B_\ell] \cdot H_f = B_f$
- 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>

$\xrightarrow{\quad x \quad}$

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

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

$\xleftarrow{\quad vk \quad}$

$\xleftarrow{\quad \sigma_x \quad}$

$\xrightarrow{\quad f,\ y,\ \sigma_{f,y} \quad}$

$\downarrow$

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>.

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

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

$\xleftarrow{\quad A \quad}$

$\xrightarrow{\quad x \in \{0,1\}^\ell \quad}$

$R_1, \dots R_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times m}$

$B_i = AR_i + x_i G$

$\xleftarrow{\quad (A, B_1, \dots, B_\ell) \quad}$

$\xrightarrow{\quad f, y, R^* \quad}$

$\downarrow$

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

use $R - R^*$ as trapdoor for A to sample $A^{-1}(0)$

<u>Observe</u>: B correctly simulates verification key by LHL

suppose A succeeds: then $AR^* = B_f - y \cdot G$

$AR = B_f - f(x) \cdot G$

$\Rightarrow A(R - R^*) = \underbrace{(f(x) - y)}_{f(x) \neq y \text{ so } f(x) - y \in \{-1, 1\}} \cdot G$

R is short since signature verifies

R^* is short since R, $H_{f,x}$ are small

$\rightsquigarrow$ $R - R^*$ is a trapdoor for A

Context-hiding for homomorphic signatures:

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

Alice		Server		Bob
	$x, \sigma_x \longrightarrow$		$\longleftarrow f$	
			$y = f(x),\ \sigma_{f,y} \longrightarrow$	

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 $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$, $x \in \{0,1\}^\ell$, and $f : \{0,1\}^\ell \to \{0,1\}$:

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

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:

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

this is a function of x!

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

Suppose $\quad AR_{f,x} = B_f - y \cdot G \quad$ 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 $Vt = z$:

$$t \leftarrow V^{-1}(z) \text{ using trapdoor } \begin{bmatrix} -R_{f,x} \\ I \end{bmatrix}$$

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

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

quality of trapdoor is $\left\| \begin{bmatrix} -R_{f,x} \\ I \end{bmatrix} \right\|$, which is $(n \log q)^{O(d)}$ so norm bound is also $(n \log q)^{O(d)}$

Recap:

homomorphic encryption

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

ct: $C = AR + \mu \cdot G$

C: ciphertext; R: encryption randomness; μ: message

homomorphic signatures

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

target matrix (in vk)

signature: $AR = B - \mu\, G$

R: signature; μ: message

GSW homomorphisms are homomorphic on both messages <u>and</u> on randomness

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

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

$$\|$$

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

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

homomorphism on message

homomorphism on randomness

HE: ciphertext evaluation

HS: signature evaluation

HS: verification

Another view: We can view GSW / homomorphic signatures as homomorphic commitment scheme:

pp: $A \in \mathbb{Z}_q^{n \times m}$

to commit to a message $x \in \{0,1\}$, sample $R \xleftarrow{R} D_{\mathbb{Z},s}^{m \times t}$ and output $C \leftarrow AR + x \cdot G$

to open a commitment to message μ, reveal R and check that

$$C = AR + \mu \cdot G \quad \text{and} \quad \|R\|_\infty \leq \beta \quad \text{(for some noise bound } \beta)$$

Observe: commitment is just GSW ciphertext, so supports arbitrary computation

$$\begin{array}{l} C_1 = AR_1 + x_1 \cdot G \\ \vdots \qquad \vdots \\ C_\ell = AR_\ell + x_\ell \cdot G \end{array} \implies C_f = AR_{f,x} + f(x) \cdot G$$

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

verifier computes C_f from $C_1, \ldots, C_\ell$

can be used to open to $f(x)$

Two possible "modes": 1. Suppose A is an LWE matrix: $A = \begin{bmatrix} \bar{A} \\ \bar{s}^T \bar{A} + e^T \end{bmatrix}$.

Then, the commitment scheme is extractable: given trapdoor information, can extract unique message for which an opening exists (if there is such a message).

If C can be opened to $\mu \in \{0,1\}$, then there exists short R such that

$$C = AR + \mu \cdot G \implies s^T C = s^T AR + \mu \cdot s^T G \qquad (s = [-\bar{s} \mid 1])$$

$$= e^T R + \mu \cdot s^T G$$

$$\approx \mu \cdot s^T G \quad \text{which suffices to recover } \mu$$

Extractable commitment $\implies$ statistically binding

2. Suppose A is random matrix: $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$

Then, the commitment scheme is equivocable: given trapdoor information, can open a commitment to both 0 or 1.

To see this, sample $(A, T) \leftarrow \text{TrapGen}(n, q)$. Then A is statistically close to uniform.

To generate opening for commitment C to message $\mu \in \{0,1\}$,

$$R \leftarrow \text{SamplePre}(A, T, C - \mu G, s)$$

This yields short R where

$$AR = C - \mu G \implies C = AR + \mu \cdot G$$

Equivocable commitment $\implies$ statistically hiding

Succinct homomorphic commitments (i.e., functional commitments):

Commitment to x: $\left.\begin{array}{l} C_1 = AR_1 + x_1 G \\ \quad \vdots \\ C_\ell = AR_\ell + x_\ell G \end{array}\right\}$ grows with the input length ℓ

Can we compress further? Yes, but will need a stronger assumption.

ℓ-succinct SIS: SIS with respect to $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ holds even given a trapdoor for the related matrix

$$B = \begin{bmatrix} A & & & W_1 \\ & A & & W_2 \\ & & \ddots & \vdots \\ & & A & W_\ell \end{bmatrix} \quad \text{where } W_i \xleftarrow{R} \mathbb{Z}_q^{n \times t}$$

Note: When W_i's are very wide ($t \sim \Omega(\ell n \log q)$), then SIS $\implies$ ℓ-succinct SIS [challenge problem]

For succinct commitments, we will set $t = m$.