# Next up: homomorphic signatures

**client** | **server**

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

$\xrightarrow{\quad x,\ \sigma \quad}$

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

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

$\downarrow$

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 $\sigma$ 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 $f$ ↓ $d$)
- $\text{Verify}(vk_f, y, \sigma) \to 0/1$ — Runs in time $\text{poly}(\lambda, d, |y|)$

Homomorphic signatures allow computations on authenticated data.

**Defining unforgeability:** adversary | challenger

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

$\xleftarrow{\quad vk \quad}$

$\xrightarrow{\quad x \quad}$

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

One-time security (generalizes to many-time)

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

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

$\downarrow$

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

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

- $\text{KeyGen}(1^\lambda)$: Set lattice parameters $n = n(\lambda)$, $q = q(\lambda)$. Let $s = s(\lambda)$ be Gaussian width parameter for preimage sampling.
  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 $B_1, \dots, B_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times t}$
  Output $vk = (A, B_1, \dots, B_\ell)$, $sk = T$
- $\text{Sign}(sk, x)$: Compute $R_i \leftarrow \text{SamplePre}(A, T, B_i - x_i G)$ for $i \in [\ell]$
  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 $\sigma = (R_1, \dots, R_\ell)$
- $\text{Verify}(vk, x, \sigma)$: Check that $\|R_i\| \leq B$ where $B = s \cdot \omega(\sqrt{\log n})$ and that $A[R_1 | \cdots | R_\ell] \stackrel{?}{=} [B_1 | \cdots | B_\ell] - x \otimes G$

**Homomorphic evaluation:** $A[\underbrace{R_1|\cdots|R_\ell}_{\text{signatures}}] = [\underbrace{B_1 - x_1G|\cdots|B_\ell - x_\ell G}_{\text{verification keys}}]$

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

$$\left.\begin{aligned} R_+ &= R_i + R_j \\ B_+ &= B_i + B_j \end{aligned}\right\} \text{Verification: } AR_+ \overset{?}{=} B_+ - (x_i + x_j)G$$

new verification component associated with addition operation

new signature

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

$$AR_i = B_i - x_iG \qquad \Rightarrow \text{ desire something of the form}$$
$$AR_j = B_j - x_jG \qquad\qquad AR_\times = B_\times - x_ix_j \cdot G$$

function of $R_i, R_j$ and $x_i, x_j$ (should be short)

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

$$AR_i = B_i - x_iG \longrightarrow B_i = AR_i + x_iG$$
$$\begin{aligned} AR_jG^{-1}(B_i) &= (B_j - x_j \cdot G)G^{-1}(B_i) \\ &= B_jG^{-1}(B_i) - x_jB_i \\ &= B_jG^{-1}(B_i) - A(x_jR_i) - x_ix_jG \end{aligned}$$
$$\Rightarrow A\underbrace{(R_jG^{-1}(B_i) + x_jR_i)} = \underbrace{B_jG^{-1}(B_i)} - x_ix_j \cdot G$$

$$R_\times = R_jG^{-1}(B_i) + x_jR_i \qquad B_\times = B_jG^{-1}(B_i)$$

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

function of public key only (this is GSW homomorphic multiplication)

**Observation:**
$$R_+ = R_i + R_j = [R_i | R_j]\begin{bmatrix} I_t \\ I_t \end{bmatrix}$$
$$R_\times = R_i(x_jI_t) + R_jG^{-1}(R_i) = [R_i | R_j]\begin{bmatrix} x_jI_t \\ G^{-1}(R_i) \end{bmatrix}$$

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 $d$, $\|R_{f,x}\|_\infty \leq t^{O(d)}$

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 $\sim t^{O(d)}$)

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

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

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

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

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

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

This can be expanded as
$$\begin{aligned} AR_{f,x} = A[R_1|\cdots|R_\ell]H_{f,x} &= [B_1 - x_1G|\cdots|B_\ell - x_\ell \cdot G]H_{f,x} \\ &= B_f - f(x) \cdot G \end{aligned}$$