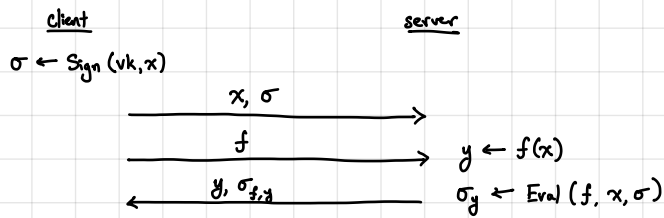


## Next up: homomorphic signatures



↓  
 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!

- Preprocess  $(vk, f) \rightarrow vk_f$
- Verify  $(vk_f, y, \sigma) \rightarrow 0/1$

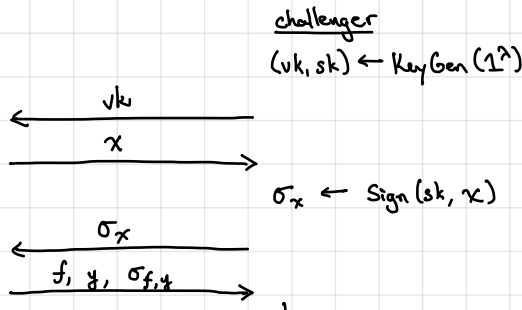
Generates short function verification key  $vk_f$  ( $|vk_f| = \text{poly}(\lambda, d)$ )  
 Runs in time  $\text{poly}(\lambda, d, |y|)$

depth of circuit  
 computing  $f$   
 ↓

Homomorphic signatures allow computations on authenticated data.

Defining unforgeability: adversary

One-time security  
 (generalizes to many-time)



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

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

- KeyGen  $(1^\lambda)$ : Set lattice parameters  $n = n(\lambda)$ ,  $q = q(\lambda)$ .

Sample  $(A, T) \leftarrow \text{TrapGen}(n, q)$  [ $A \in \mathbb{Z}_q^{n \times m}$ ,  $T \in \{0,1\}^{m \times t}$ ]

Sample  $B_1, \dots, B_\ell \xleftarrow{R} \mathbb{Z}_q^{n \times t}$

Output  $vk = (A, B_1, \dots, B_\ell)$ ,  $sk = R$

↳  $AT = G \in \mathbb{Z}_q^{n \times t}$ ;  $t = n \lceil \log q \rceil$

- Sign  $(sk, x)$ : Compute  $R_i \leftarrow A^{-1}(B_i - x_i G)$  for  $i \in [\ell]$  using  $T$

In particular:

$$A[R_1 \mid \dots \mid R_\ell] = [B_1 - x_1 G \mid \dots \mid B_\ell - x_\ell G] \quad (R_i \in \mathbb{Z}_q^{n \times t})$$

$$= [B_1 \mid \dots \mid B_\ell] - x \otimes G$$

Output  $\sigma = (R_1, \dots, R_\ell)$

- Verify  $(vk, x, \sigma)$ : Check that  $\|R_i\| \leq B$  and that  $A[R_1 \mid \dots \mid R_\ell] \stackrel{?}{=} [B_1 \mid \dots \mid B_\ell] - x \otimes G$

↳ bound based on quality of trapdoor (lattice parameters)

Homomorphic evaluation:  $A[R_1 | \dots | R_\ell] = [B_1 - x_1 G | \dots | B_\ell - x_\ell G]$

To derive a signature on the sum of two bits  $(x_i + x_j)$ :  
 $R_+ = R_i + R_j$   
 $B_+ = B_i + B_j$   
 Verification:  $AR_+ \stackrel{?}{=} B_+ - (x_i + x_j)G$   
 (Annotations: "new verification component associated with addition operation", "new signature")

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

$AR_i = B_i - x_i G \Rightarrow$  desire something of the form  
 $AR_j = B_j - x_j G$   
 $AR_x = B_x - x_i x_j G$

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  $x$ )

$\rightarrow AR_i = B_i - x_i G \rightarrow B_i = AR_i + x_i G$   
 $AR_j G^{-1}(B_i) = (B_j - x_j G) G^{-1}(B_i)$   
 $= B_j G^{-1}(B_i) - x_j B_i$   
 $= B_j G^{-1}(B_i) - A(x_j R_i) - x_i x_j G$

$\Rightarrow A(R_j G^{-1}(B_i) + x_j R_i) = B_j G^{-1}(B_i) - x_i x_j G$

$R_x = R_j G^{-1}(B_i) + x_j R_i$      $B_x = B_j G^{-1}(B_i)$

function of signature, input

$\|R_x\|_\infty \leq \|R_j\|_\infty \cdot t + \|R_i\|_\infty$

function of public key only

(this is GSW homomorphic multiplication)

Observation:  $R_+ = R_i + R_j$   
 $R_x = R_i(x_j I_t) + R_j G^{-1}(R_i)$   
 $= [R_i | R_j] \begin{bmatrix} I_t \\ I_t \end{bmatrix}$   
 $= [R_i | R_j] \begin{bmatrix} x_j I_t \\ G^{-1}(R_i) \end{bmatrix}$   
 (Annotations: "can depend on  $R_i, R_j, x$ ", "small linear function of  $R_i$  and  $R_j$ ", " $R_+$ ", " $R_x$ ")

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

this can also be written as

$B_f \leftarrow [B_1 | \dots | B_\ell] \cdot H_f$  where  $\|H_f\| \leq m^{O(d)}$

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

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

More generally:

$R_{f,x} = [R_1 | \dots | R_\ell] \cdot H_{f,x}$  where  $H_{f,x} \in \mathbb{Z}_q^{t \times t}$  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  $AR_i = B_i - x_i G$ , then from the above,

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

where  $B_f$  is the matrix obtained by evaluating  $f$  on  $B_1, \dots, B_\ell$

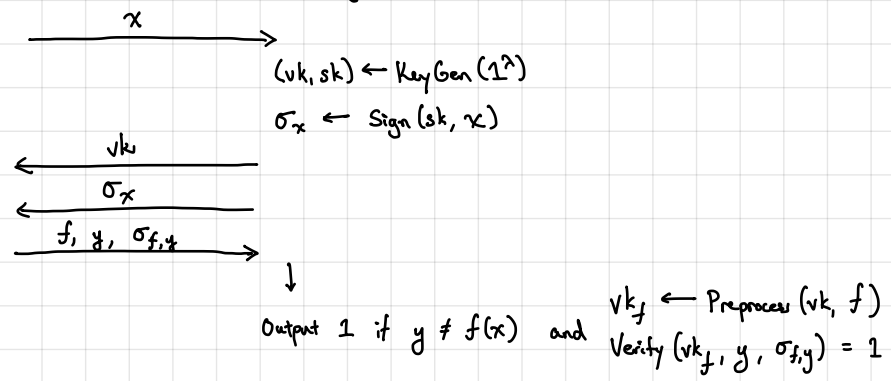
This can be expanded as

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

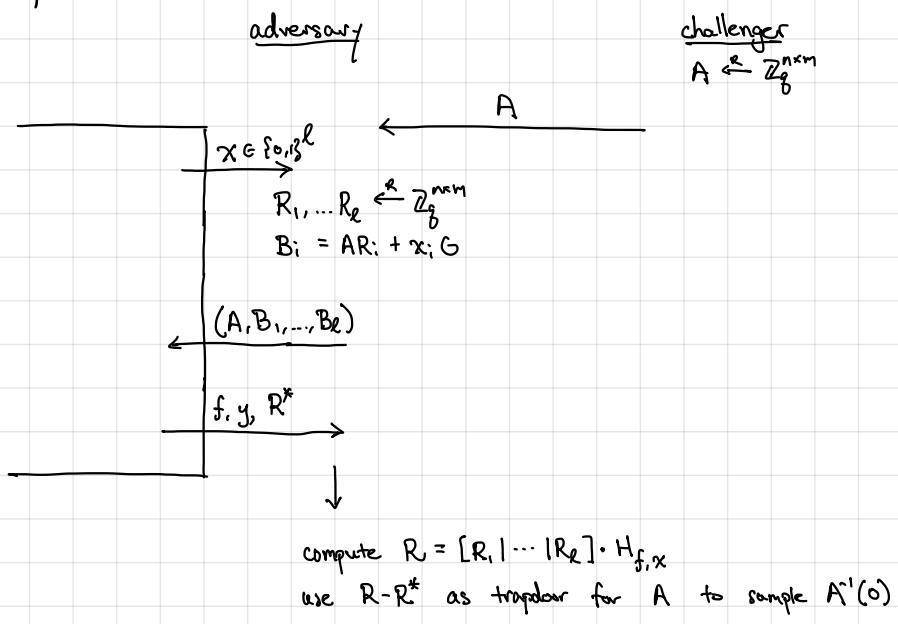
Decouple into two equations:

- Input-independent evaluation:  $[B_1 | \dots | B_\ell] \cdot H_f = B_f$
  - Input-dependent evaluation:  $[B_1 - x_1 G | \dots | B_\ell - x_\ell G] H_{f,x} = B_f - f(x) \cdot G$
- ] Will give us many advanced primitives!

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



Proof of unforgeability.



Observe: B correctly simulates verification key by LHL

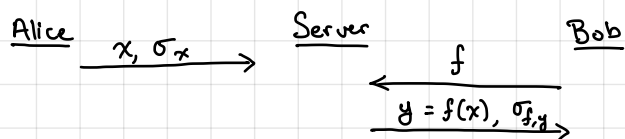
suppose A succeeds: then  $AR^* = B_f - y \cdot G \Rightarrow A(R - R^*) = \underbrace{(f(x) - y)}_{f(x) \neq y \text{ so } f(x) - y \in \mathbb{F}^{-1,1}} \cdot G$

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

$R$  is short since signature verifies  $\leadsto R - R^*$  is a trapdoor for  $A$   
 $R^*$  is short since  $R, H_{f,x}$  are small

Context-hiding for homomorphic signatures:

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



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 <sup>statistically</sup> context-hiding if there exists an efficient simulator  $S$  where for all  $(vk, sk) \leftarrow \text{KeyGen}(1^\lambda)$ ,  $x \in \{0,1\}^l$ , and  $f: \{0,1\}^l \rightarrow \{0,1\}$ :

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

↳ 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  $f(x)$

Turns out this is not difficult to achieve!

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

Current construction is not context-hiding:

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

↳ 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  $V$ :

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

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  $Vt = z$ :

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

To verify a signature, the verifier computes  $B_f$  from  $B_1, \dots, B_l$ , constructs  $V$  from the verification key and checks that  $Vt = z$  and  $\|t\|_\infty \leq \beta$  where  $\beta = (n \log q)^{O(d)}$  is the noise bound

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

Recap: homomorphic encryption

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

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

↑ ciphertext    ↑ encryption randomness    ← message

homomorphic signatures

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

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

target matrix (in vk)    ← message  
↑ signature

GSW homomorphisms are homomorphic on both messages and on randomness

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

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

$$\parallel$$

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

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

homomorphism on message  
homomorphism on randomness

HE: ciphertext evaluation

HS: verification

HS: signature evaluation

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  $R \xleftarrow{R} \mathbb{Z}_q^{m \times t}$  and output  $C \leftarrow AR + x \cdot G$

to open a commitment to message  $\mu$ , reveal  $R$  and check that

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

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

$$C_1 = AR_1 + x_1 \cdot G$$

$\vdots$

$$C_\ell = AR_\ell + x_\ell \cdot G$$

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

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

verifier computes

$C_f$  from  $C_1, \dots, C_\ell$

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

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

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

$$\begin{aligned} C = AR + \mu \cdot G &\Rightarrow s^T C = s^T AR + \mu \cdot s^T G & (s = [-\bar{s} \quad | \quad 1]) \\ &= e^T R + \mu \cdot s^T G \\ &\approx \mu \cdot s^T G \text{ which suffices to recover } \mu \end{aligned}$$

Extractable commitment  $\Rightarrow$  statistically binding

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

Then, the commitment scheme is equivocal: 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

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

Equivocal commitment  $\Rightarrow$  statistically hiding

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

Commitment to  $x$ :  $C_1 = AR_1 + x_1 \cdot G$

$\vdots$

$C_\ell = AR_\ell + x_\ell \cdot G$

} grows with the input length  $\ell$

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

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

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

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

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