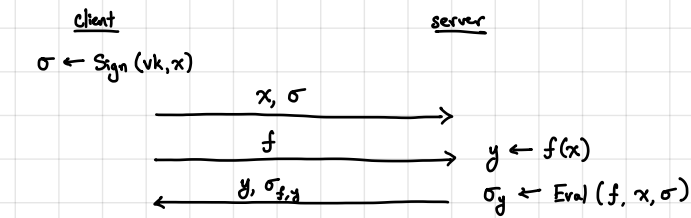


Next up: homomorphic signatures



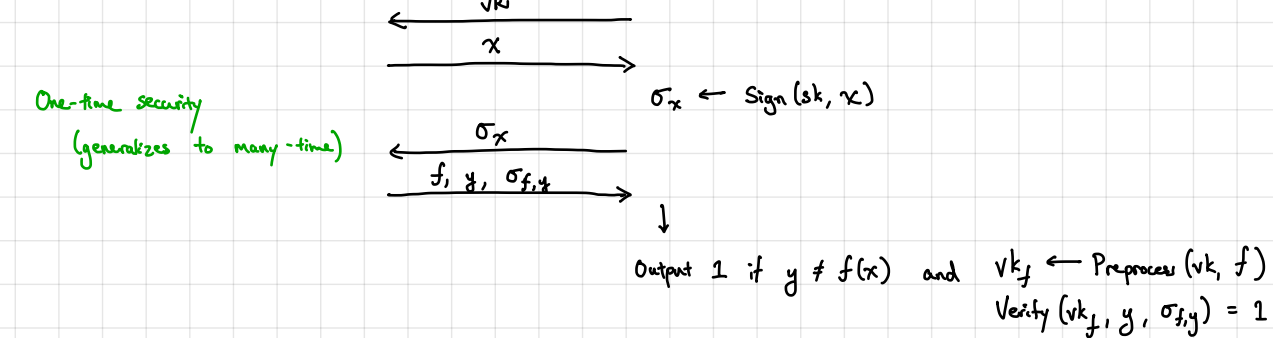
↓  
 checks that  $\sigma_{f,y}$  is a signature on  $y$  with respect to function  $f$   
 ↪ can view as signature on pair  $(f, y)$  ← Why not just on  $y$  alone?

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  $(vk, f) \rightarrow vk_f$  Generates short function verification key  $vk_f$  ( $|vk_f| = \text{poly}(\lambda, d)$ )  
 - Verify  $(vk_f, y, \sigma) \rightarrow 0/1$  Runs in time  $\text{poly}(\lambda, d, |y|)$

depth of circuit computing  $f$   
 ↓

Homomorphic signatures allow computations on authenticated data.



One-time security  
 (generalizes to many-time)

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

- $\text{KeyGen}(1^\lambda)$ : Set lattice parameters  $n = n(\lambda)$ ,  $q = q(\lambda)$ . Let  $s = s(\lambda)$  be Gaussian width parameter for preimage sampling.  
 Sample  $(A, T) \leftarrow \text{TrapGen}(n, q)$  [ $A \in \mathbb{Z}_q^{n \times m}$ ,  $T \in \{0,1\}^{m \times t}$ ]  
 Sample  $B_1, \dots, B_\ell \xleftarrow{r} \mathbb{Z}_q^{n \times t}$  ↳  $AT = G \in \mathbb{Z}_q^{n \times t}$ ;  $t = n \lceil \log q \rceil$   
 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:  
 $A[R_1 | \dots | R_\ell] = [B_1 - x_1 G | \dots | B_\ell - x_\ell G]$  ( $R_i \in \mathbb{Z}_q^{n \times t}$ )  
 $= [B_1 | \dots | B_\ell] - x \otimes G$   
 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  $A[R_1 | \dots | R_\ell] \stackrel{?}{=} [B_1 | \dots | B_\ell] - x \otimes G$

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  
 $AR_j = B_j - x_j G$   
 $AR_x = B_x - x_i x_j G$

(Annotations: "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  $x$ )")

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

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

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

(Annotations: "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  $R_j$ ", " $R_x$ ")

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  $B_1, \dots, B_\ell$  and checks that  $\|R_{f,x}\|_\infty$  sufficiently small (bound  $\sim t^{O(d)}$ )  
 $AR_{f,x} = B_f - z \cdot G$

More generally:

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

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