Next up: homomorphic signatures

\[
\begin{align*}
\sigma & \leftarrow \text{Sign}(vk, x) \\
\sigma_y & \leftarrow \text{Eval}(f, \sigma) \\
\end{align*}
\]

\[\n\downarrow\]

Checks that \(\sigma_y\) is a signature on \(y\) with respect to function \(f\).

Requirements:

- Unforgeability: Cannot construct signature \(\sigma\) on \((f, y)\) where \(y \neq f(x)\).
- Succinctness: Size of \(\sigma_y\) should be \(|f| \cdot \text{poly}(\lambda)\). In particular, should not depend on \(|x|\) or \(|f|\).
- Efficient verification: Can decompose verification algorithm as follows:
  - \(\text{Preprocess}(vk, f) \rightarrow vk_f\)
  - \(\text{Verify}(vk_f, y, \sigma_y) \rightarrow 0/1\)

Homomorphic signatures allow computations on authenticated data.

**Defining unforgeability:** adversary

\[\begin{align*}
\text{challenger} & \quad (vk, sk) \leftarrow \text{KeyGen}(\lambda^2) \\
\sigma_x & \leftarrow \text{Sign}(sk, x) \\
\end{align*}\]

**One-time security** (generalize to many-time)

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

\[\downarrow\]

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

\[\text{Verify}(vk_f, y, \sigma_y) = 1\]

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

- \(\text{KeyGen}(\lambda^2)\):
  - Set lattice parameters \(\lambda = \lambda(n), g \cdot \lambda(x)\). Let \(s \cdot \lambda(x)\) be Gaussian width parameter for preimage sampling.
  - Sample \((A, T) \leftarrow \text{TrapGen}(n, g)\) \[A \in \mathbb{Z}_g^{m \times t}, T \in \text{poly}(\lambda^2)\]
  - Sample \(B_1, \ldots, B_k \in \mathbb{Z}_g^{m \times t}\)
  - Output \(vk = (A, B_1, \ldots, B_k), sk = T\)

- \(\text{Sign}(sk, x)\):
  - Compute \(R_i \leftarrow \text{SampleRe}(\text{pre}(A, T, B_i, -x; G)\) for \(i \in [k]\)
  - In particular:
    - \(A[R_1, \ldots, R_k] = [B_1, -x, G] \cdots [B_k, -x, G]\)
    - \(R_i \in \mathbb{Z}_g^{m \times t}\)
    - \(R_i = [B_1, \ldots, B_k] - x \otimes G\)
  - Output \(\sigma = (R_1, \ldots, R_k)\)

- \(\text{Verify}(vk, x, \sigma)\):
  - Check that \(\|R_i\| = B\) when \(B := s \cdot \lambda(\sigma_y)\) and that \(A[R_1, \ldots, R_k] = [B_1, \ldots, B_k] - x \otimes G\)
Homomorphic evaluation: \[ A \left[ R_1, \ldots, R_k \right] = [B, -x, G \mid \ldots \mid B, -x_k, G] \]

To derive a signature on the sum of two bits \( (x_i + x_j) \):
\[
\begin{align*}
R_x &= R_i + R_j \\
B_x &= B_i + B_j
\end{align*}
\]
function of \( R_i, R_j \) should be short (signature algorithm does not know \( x \))

To derive a signature on the product of two bits \( (x_i; x_j) \):
\[
\begin{align*}
AR_i &= B_i - x_i, G \\
AR_j &= B_j - x_j, G \\
AR_x &= B_x - x_i x_j, G
\end{align*}
\]
function of \( x_i, x_j \) should not depend on \( B_i, B_j \) - short key only

\[
\begin{align*}
R_x &= R_j G'(B_i) + x_j, B_i \Rightarrow B_x &= B_j G'(B_i)
\end{align*}
\]
\[ R_x \text{ is general key, input} \quad \| R_x \|_{\infty} \leq \| R_i \|_\infty \quad \text{(Arb is given homomorphic multiplication)} \]

Small linear function of \( R_i \) and \( R_j \)

Composing above operations to compute signature on \( R^x \) on evaluation \( f(x) \)

By above analysis, multiplication scales noise by a factor of \( t \) so if \( t \) can be computed by a circuit of depth \( d' \), \( \| R^x \|_{\infty} \leq t^{\log d'} \)

To verify a signature \( R^x \) on \( (f; z = f(x)) \), verifier computes \( B_j \) from \( B_1, \ldots, B_k \) and checks that
\[ R^x = B_j - z, G \]

More generally
\[ R^x = \left[ R_1, \ldots | R_k \right] . H_f, x \quad \text{where} \quad H_f, x \in \mathbb{Z}_q^{-2^{k+1}} \quad \text{and} \quad \| R^x \|_{\infty} \leq t^{\alpha} = (n \log q)^{\alpha} \]

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

Now, if \( AR_i = B_i - x_i, G \), then from the above,
\[ AR^x = B_j - f(x), G \]

This can be expanded as
\[ AR^x = A \left[ R_1, \ldots, R_k \right] H_f, x = [B - x, G \mid \ldots \mid B, -x_k, G] H_f, x = \bar{B}_f - f(x), G \]