2. Hash-Based Signatures

Lamport signature scheme

Algorithm

Let

- $M \in \{0,1\}^v$ be a message. We split it in bits => $M = m_0 || m_1 || ...$
- $f : \mathcal{X} \to \mathcal{Y}$ be a one way function (a hash for example)

Key generation

- for each bit m the signer chooses 2 values
  - $(x_0, x_1) \to$ **Private key**
  - $(y_0, y_1) = (f(x_0), f(x_1)) \to$ **Public key**

Signing $m \in \{0,1\}$

- $\sigma = S((x_0, x_1), m) = x_m$
- If the bit is 0 take the x_0 else take y_1
- Sign each bit accordingly

Verifying $m \in \{0,1\}$

- $V((y_0, y_1), m, \sigma) = accept \iff f(\sigma) = y_m \; else \; reject$
- Check for all bits

Note

- The size of the keys is $2 * v * bitsize(x), 2 * v * bitsize(y)$
- To make it more efficient we can use a PRNG / PRF and keep the seed as the secret key
- There are way to shorten the signatures
  - Winternitz scheme
  - HORS
  - using Merkle trees

Security

The Lampert signature is a one time signature.

- If a key is reused an attacker can queue $m_0 = 0^v, m_1 = 1^v$ and find the secret key then he can forge a signature for a message $m' \neq m_0, m' \neq m_1$

    - https://crypto.stackexchange.com/questions/2640/lamport-signature-how-many-signatures-are-needed-to-forge-a-signature
- If the key is not reused only half of the secret key is known and the attacker must brute force / reverse a the one way f to forge a signature
- Therefore the security is based on the key length and the one way resistance of f

Resources

- http://lamport.azurewebsites.net/pubs/dig-sig.pdf
- https://en.wikipedia.org/wiki/Lamport_signature
- https://www.youtube.com/watch?v=be3DJEaOYfg
- https://www.youtube.com/watch?v=v3yApr-8Ilo