Lecture 19

- Lamport's one-time sigs (again)
- Merkle trees
- GHR signatures

Lamport's one-time sigs (again)

$$\text{OWF } f$$

Message space: $$\{0,1\}^n$$

$$\text{Sign}(SK, m): \{x_{m_i}, 0\}^n$$ where $$m_i$$ is the $$i$$th bit of $$m$$

$$\text{Verify}(PK, m, s): \text{check that } \forall i, f(s_i) = PK_{m_i}$$

One-time security: A's signing oracle only answers the query.

Reduction:

$$y = f(x)$$

$$x' \text{ st. } f(x') = y$$

Reduction: Compute $$PK$$: Pick random $$i, b$$, let $$PK_{b, i} = y$$

for $$(b', i') \neq (b, i)$$, pick $$x_{b', i'} \leftarrow \{0,1\}^n$$

$$PK_{b', i'} = f(x_{b', i'})$$

Signing query: $$m$$ if $$m_i = b$$, then fail
else output 6 correctly

Processing forgery ($$m', 6'$$): if $$m_i \neq b$$, then fail
else, output 6'

Analysis of the reduction:

Intermediate experiment $$B'$$: Compute $$PK, SK$$ correctly. When faced with sig query
from $$A$$, $$B'$$ chooses $$i, b$$ and fails if $$m_i = b$$.

When $$A$$ produces $$(m', 6')$$, $$B'$$ fails if $$m_i \neq b$$
else, if $$6'$$ is a successful forgery
then output "success."
Intermediate exp #2. B" : B" computes PRISK correctly
Always answer the Sig query
pick i, b, afterward it determines success or failure.

B" interacting with u(A^w):
1. PK scheme as wild
2. Sig query always answered (m, e)
3. Forgery (m', e')
successful if: w with prob \( \frac{e(c)}{2n} \)
   1. Forgery successful w.p. e(c)
   2. \( m_i = b \) and \( m'_i = b \) for random i, b
      (i.e. \( m \) differs from \( m' \) in pos i (for random i) \( \geq \frac{1}{n} \)
      and \( m'_i = b \) (for random b) \( \frac{1}{2} \))

B': same as B" but fails faster
1. (i, b) chosen
2. PK same as in wild
3. Sig query m only answered
   if \( m_i \neq b \)
4. Forgery (m', e')
   success if forgery successful and
   \( m_i \neq m'_i \)
   success prob.
   \( \frac{e(c)}{2n} \)

Reduction B(y):
1. (i, b) chosen
2. PK chosen same as in wild
   except \( PK_{bi} = y \)
3. Sig query only answered if \( m_i \neq b \)
4. Forgery (m', e') success if forgery
   successful and \( m_i \neq m'_i \)
   success w. prob
   \( \frac{e(c)}{2n} \)

Question: Once you have a signature scheme that works only for the signing of one message how can you use it to sign many messages?

One solution is the use of Merkle Trees

One PK at a one-time sig
KeyGen: generate SK₀ ..., SKₖ using a OTS
PK₀ ... PKₖ

Let hₑ be the root of the Merkle tree where PKᵢ's are leaves.
PK = hₑ, SK = (SK₀, ..., SKₖ)

Sign (counter, SK, m):
  \( s' \leftarrow \text{OTS} \) Sign (SK, counter, m)
  counter \( \leftarrow \) counter + 1
  output (s', counter, \{hₚb \}_{ₚ \in \text{prefix of counter}})

Verify: verify that PK is in the tree
  verify s' using OTS

RSA assumption: given \((n, e, x)\), hard to find \(z\) s.t. \(z^e = x \mod n\)

Strong RSA assumption: given \((n, x)\), hard to find \((e, z)\) s.t. \(e > 1, z^e = x \mod n\)

Constructing a signature scheme:

TRY 1
PK = (n, x)
SK = n's factorization

Sign(SK, m) output \(z\) s.t. \(z^m = x \mod n\)

Let \(m = 42 = 7 \times 3 \times 2\) so after querying gets \(5^2 = x\)
so automatically get signs for \(m = 7, m = 3, m = 2\)

TRY 2
Message space consists of primes

Sign(SK, m): first check if m is prime

TRY 3
GHR
PK = (n, x, H)
SK = n's factor. H: \(\mathbb{G}_0, \mathbb{S}_x\) → primes of length k

Sign(SK, m): output \(z\) s.t. \(z^{H(m)} = x \mod n\)