

# Authentication

CS 600.442 Modern Cryptography

Fall 2016

# The Setting

- Alice wants to send a message  $m$  to Bob in such a manner that upon receipt, Bob can determine whether the message arrived untampered or not
- Want: Digital analogue of physical signatures
- Alice (“signer”) signs a message  $m$  to produce a signature  $\sigma$
- Bob (“verifier”) can verify that  $\sigma$  is indeed generated for  $m$
- Adversary cannot *forge* a signature

# Two Types

- 1 Private Key: Message Authentication Codes
- 2 Public Key: Digital Signatures

# Message Authentication Code (MAC)

- Signer and Verifier “share a secret”
- **Key Generation:**  $\text{Gen}(1^n)$  outputs secret key  $k$
- **Sign:**  $\text{Tag}_k(m)$  outputs a tag  $\sigma$
- **Verify:**  $\text{Ver}_k(m, \sigma)$  is 1 if and only if  $\sigma$  is a valid tag of  $m$  under the secret key  $k$

Security: An adversary can observe multiple (message,tag) pairs of its choice, but still cannot forge a tag on a new message

# MAC: Algorithms

- $k \leftarrow \text{Gen}(1^n)$
- $\sigma \leftarrow \text{Tag}_k(m)$
- $\text{Ver}_k: \mathcal{M} \times \mathcal{T} \rightarrow \{0, 1\}$
- Correctness:  $\Pr[k \leftarrow \text{Gen}(1^n), \sigma \leftarrow \text{Tag}_k(m): \text{Ver}_k(m, \sigma) = 1] = 1$
- Security (UF-CMA): For all n.u. PPT adversary  $\mathcal{A}$  there exists a negligible  $\nu(\cdot)$  such that:

$$\Pr \left[ \begin{array}{c} k \leftarrow \text{Gen}(1^n) \\ (m, \sigma) \leftarrow \mathcal{A}^{\text{Tag}_k(\cdot)}(1^n) \end{array} : \begin{array}{c} \mathcal{A} \text{ did not query } m \wedge \\ \text{Ver}_k(m, \sigma) = 1 \end{array} \right] \leq \nu(n)$$

## Theorem

$PRF \implies MAC$

- $\text{Gen}(1^n)$ : Output  $k \xleftarrow{\$} \{0, 1\}^n$
- $\text{Tag}_k(m)$ : Output  $f_k(m)$
- $\text{Ver}_k(m, \sigma)$ : Output  $f_k(m) \stackrel{?}{=} \sigma$
- Think: Proof?

# One-time MAC

- Weaker Security: Adversary is allowed only one query
- Advantage: Unconditional security!
- Analogue of OTP for authentication
- Think & Read

# Digital Signature

- Only Signer can sign but everyone can verify
- **Key Generation:**  $(sk, pk) \leftarrow \text{Gen}(1^n)$
- **Sign:**  $\sigma \leftarrow \text{Sign}_{sk}(m)$
- **Verify:**  $\text{Ver}_{pk}(m, \sigma): \mathcal{M} \times \mathcal{S} \rightarrow \{0, 1\}$
- Correctness:

$$\Pr[(sk, pk) \leftarrow \text{Gen}(1^n), \sigma \leftarrow \text{Sign}_{sk}(m) : \text{Ver}_{pk}(m, \sigma) = 1] = 1$$

- Security (UF-CMA):

$$\Pr \left[ \begin{array}{l} (sk, pk) \leftarrow \text{Gen}(1^n) \\ (m, \sigma) \leftarrow \mathcal{A}^{\text{Sign}_{sk}(\cdot)}(1^n, pk) \end{array} : \begin{array}{l} \mathcal{A} \text{ did not query } m \wedge \\ \text{Ver}_{pk}(m, \sigma) = 1 \end{array} \right] \leq \nu(n)$$

- One-time Signatures: Adversary is allowed only one query



# One-time Signature: Construction [Lamport]

Let  $f$  be a one-way function

- $sk := \begin{pmatrix} x_1^0 & x_2^0 & \dots & x_n^0 \\ x_1^1 & x_2^1 & \dots & x_n^1 \end{pmatrix}$ , where  $x_i^b \xleftarrow{\$} \{0, 1\}^n$  for all  $i \in [n]$  and  $b \in \{0, 1\}$
- $pk := \begin{pmatrix} y_1^0 & y_2^0 & \dots & y_n^0 \\ y_1^1 & y_2^1 & \dots & y_n^1 \end{pmatrix}$ , where  $y_i^b = f(x_i^b)$  for all  $i \in [n]$  and  $b \in \{0, 1\}$
- $\text{Sign}_{sk}(m) : \sigma := (x_1^{m_1}, x_2^{m_2}, \dots, x_n^{m_n})$
- $\text{Ver}_{pk}(m, \sigma) : \bigwedge_{i \in [n]} f(\sigma_i) \stackrel{?}{=} y_i^{m_i}$
- Think: Proof?

Think: How to sign long messages?

# Collision-resistant Hash Functions

- Intuition: A compressing function  $h$  for which it is hard to find  $x, x'$  s.t.  $x \neq x'$  but  $h(x) = h(x')$
- Impossible for non-uniform adversary notion
  - Think: Why?
- Need to consider a family of hash functions

# Collision-resistant Hash Function Family

## Definition (Collision-resistant Hash Function Family)

A family of functions  $H = \{h_i : D_i \rightarrow R_i\}_{i \in I}$  is a collision-resistant hash function family (CRHF) if:

- **Easy to Sample:** There exists a PPT  $\text{Gen}$  s.t.:  $i \leftarrow \text{Gen}(1^n)$ ,  $i \in I$
- **Compression:**  $|R_i| < |D_i|$
- **Easy to Evaluate:** There exists a poly-time algorithm  $\text{Eval}$  s.t. given  $x \in D_i$ ,  $i \in I$ ,  $\text{Eval}(x, i) = h_i(x)$
- **Collision Resistance:** For all n.u. PPT  $\mathcal{A}$ ,  $\exists$  negligible function  $\mu(\cdot)$  s.t.

$$\Pr \left[ \begin{array}{l} i \stackrel{\$}{\leftarrow} \text{Gen}(1^n), \\ (x, x') \leftarrow \mathcal{A}(1^n, i) \end{array} : x \neq x' \wedge h_i(x) = h_i(x') \right] \leq \mu(n)$$

- One-bit compression implies arbitrary bit compression
  - Think: Proof?
  - Read: Merkle Trees
- Range cannot be too small
  - Enumeration Attacks
  - Birthday Attack
- **Existence**:
  - Unlikely to be constructed from OWF or OWP [Simon98]
  - Can be constructed from number-theoretic assumptions such as factoring, discrete log

## Remarks (contd.)

- **Weaker notion:** Universal One-way Hash Functions (UOWHF)

$$\Pr \left[ \begin{array}{l} (x, \text{state}) \leftarrow \mathcal{A}(1^n), \\ i \xleftarrow{\$} \text{Gen}(1^n), \\ x' \leftarrow \mathcal{A}(i, \text{state}) \end{array} : \begin{array}{l} x \neq x' \wedge \\ h_i(x) = h_i(x') \end{array} \right] \leq \mu(n)$$

- Can be constructed from OWF [Rompel90]
- Suffices for Digital Signatures [Naor-Yung89]
- More efficient construction  
[Haitner-Holenstein-Reingold-Vadhan-Wee10]

# One-time Signatures for Long Messages

- Let  $H = \{h_i : \{0, 1\}^* \rightarrow \{0, 1\}^n\}_{i \in I}$  be a CRHF
- Idea: Sign  $h_i(m)$  instead of  $m$  using Lamport signature
- Think: Proof?

# Multi-message Signatures (via chain)

- $(sk_0, pk_0) \xleftarrow{\$} \text{Gen}(1^n)$
- Initialize:  $\tilde{\sigma}_i = \emptyset, i = 1$
- To sign  $m_i$ :
  - $(sk_i, pk_i) \xleftarrow{\$} \text{Gen}(1^n)$
  - $\tilde{\sigma}_i \leftarrow \text{Sign}_{sk_{i-1}}(m_i \| pk_i)$
  - Output:  $\sigma_i = (i, \tilde{\sigma}_i, m_i, pk_i, \sigma_{i-1})$
  - Increment  $i$
- Think: Proof?
- Think: How to reduce signature size?
- Read: Tree-based signatures
- Read: Efficient Signatures from Trapdoor Permutations in the Random Oracle Model
- Read: Full Domain Hash (RSA-based signatures)