האוניברסיטה העברית בירושלים
The Hebrew University of Jerusalem

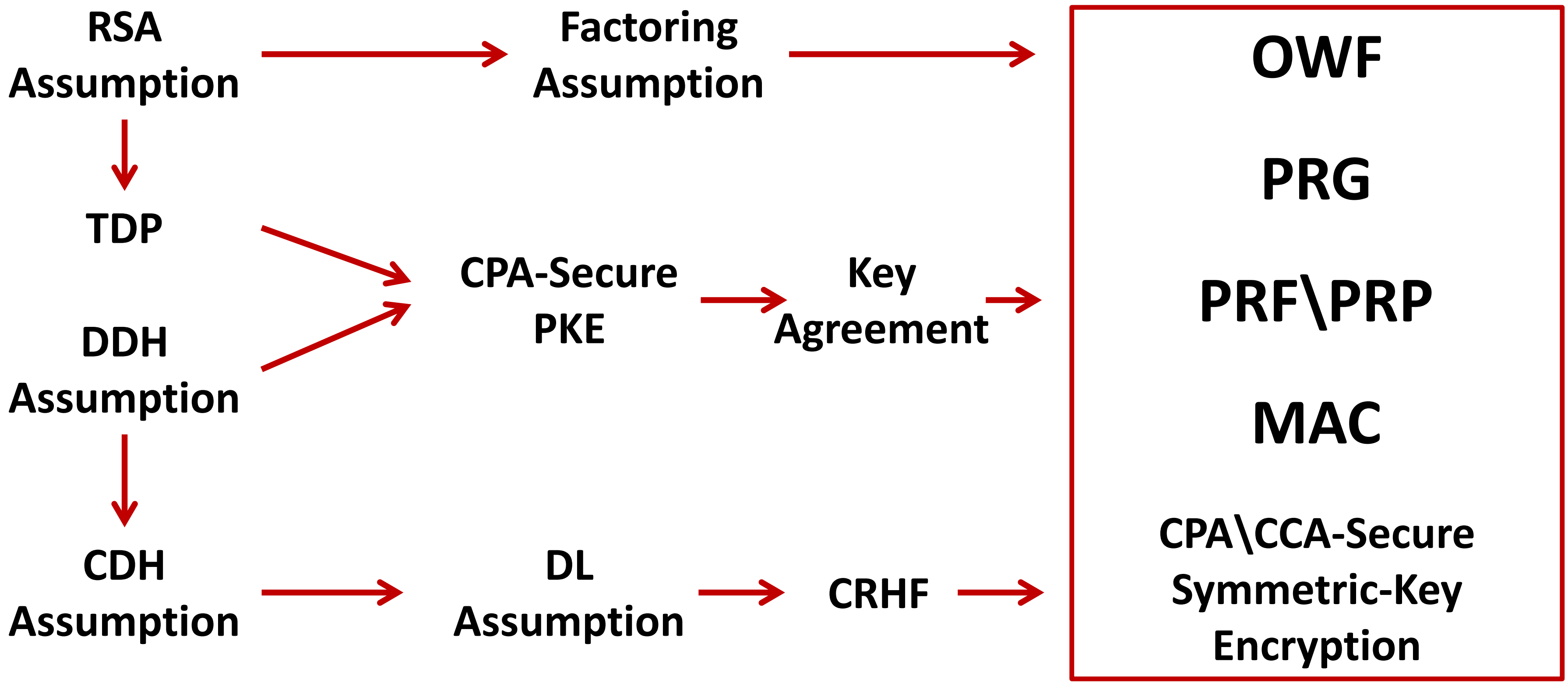
Cryptography

Lecture 10:

Digital Signatures

Gil Segev

The World of Crypto Primitives



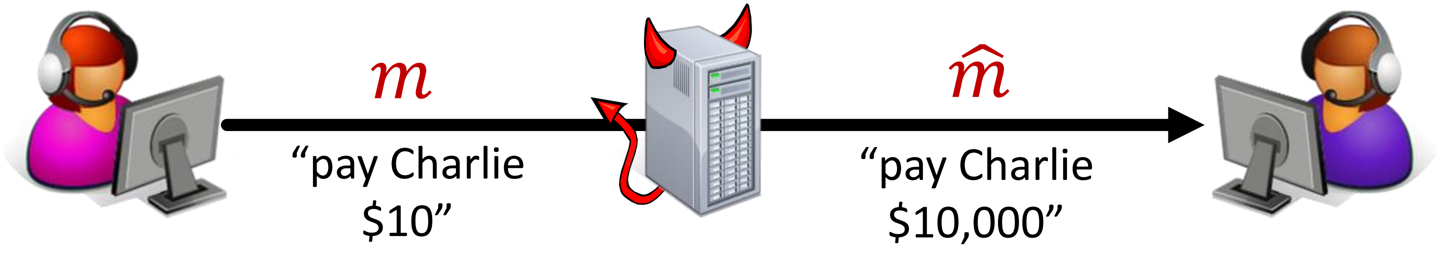
Outline

- **Digital signatures**
- **Constructions**
    - One-time signatures
    - Stateful signatures
    - Stateless signatures
- **Certificates and public-key infrastructure**
- **User-server identification**

Digital Signatures

Alice and Bob wish to communicate

- Eve completely controls the channel
- Would like to assure the receiver of a message that it has not been modified



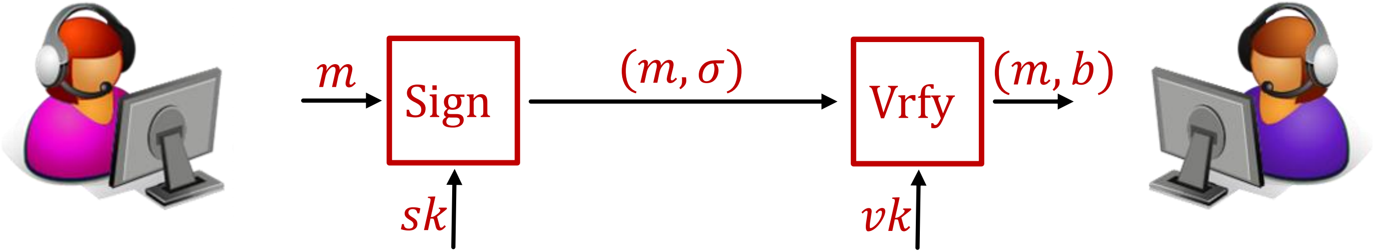
Public-key counterpart of message-authentication codes

- Signer holds a secret signing key
- Verifier knows the corresponding public verification key

Digital Signatures

Syntax: $\mathbf{\Pi = (Gen, Sign, Vrfy)}$

- Key-generation algorithm Gen on input 1^n outputs a signing key sk and a verification key vk
- Signing algorithm Sign takes a signing key sk and a message m, and outputs a signature σ
- Verification algorithm Vrfy takes a verification key vk, a message m and a signature σ, and outputs a bit b



Correctness: For every message m

$$\Pr\left[\mathrm{Vrfy}_{vk}\left(m, \mathrm{Sign}_{sk}(m)\right) = 1\right] = 1$$

Signatures vs. MACs

Signatures

- n users require only n secret keys
- Same signature can be verified by all users
- Publicly verifiable and transferable
- Provide non-repudiation

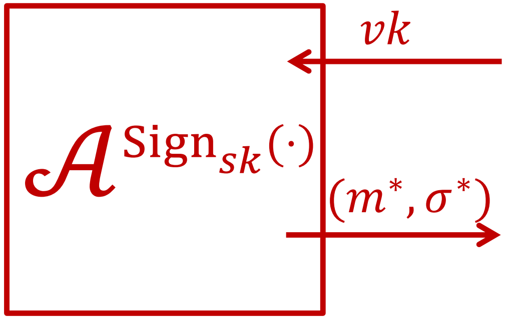
MACs

- n users require $\approx n^2$ secret keys
- Privately verifiable and non-transferable
- More efficient (2-3 orders of magnitude faster)

The Security of Signatures

- $\mathcal{A}$ knows vk and can adaptively ask for signatures of messages of its choice
- $\mathcal{A}$ tries to forge a signature on a new message

$$(sk, vk) \leftarrow \mathrm{Gen}(1^n)$$



Definition:
Π is **existentially unforgeable against an adaptive chosen-message attack** if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that

$$\Pr\left[\mathrm{SigForge}_{\Pi,\mathcal{A}}(n) = 1\right] \leq \nu(n)$$

$\mathcal{Q}$ = Set of all queries asked by $\mathcal{A}$

$$\mathrm{SigForge}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } \mathrm{Vrfy}_{vk}(m^*, \sigma^*) = 1 \\ & \text{and } m^* \notin \mathcal{Q} \\ 0, & \text{otherwise} \end{cases}$$

- ***T*-time security (weaker notion):** $\mathcal{A}$ is allowed to ask for at most T signatures

Outline

- **Digital signatures**
- **Constructions**
  - One-time signatures
  - Stateful signatures
  - Stateless signatures
- **Certificates and public-key infrastructure**
- **User-server identification**

Construction Outline

$$\boxed{\text{One-time signature scheme}}$$

$\downarrow$

Stateful signature scheme

$\downarrow$

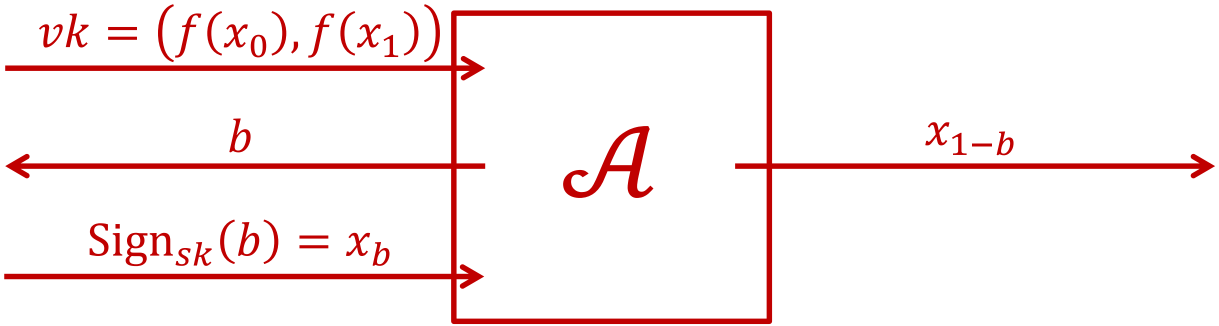
Stateless signature scheme

Lamport's One-Time Scheme

$$sk = \boxed{x_0} \boxed{x_1}$$

$$vk = \boxed{f(x_0)} \boxed{f(x_1)}$$

$$\mathrm{Sign}_{sk}(b) = x_b$$



Lamport's One-Time Scheme

Let f be a OWF. Define a signature scheme $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ for 1-bit messages as follow:

- $\mathrm{Gen}(1^n)$: Sample $x_0, x_1 \leftarrow \{0,1\}^n$ and compute $y_0 = f(x_0)$ and $y_1 = f(x_1)$. Output $sk = (x_0, x_1)$ and $vk = (y_0, y_1)$.
- $\mathrm{Sign}_{sk}(b)$: Output $\sigma = x_b$.
- $\mathrm{Vrfy}_{vk}(b, \sigma)$: If $f(\sigma) = y_b$ output 1, and otherwise output 0.

Theorem:
If f is a OWF then Π is a secure one-time signature scheme for 1-bit messages.

Proof idea:

- $\mathcal{A}$ forges a signature on b^* $\Rightarrow$ $\mathcal{A}$ inverts $y_{b^*} = f(x_{b^*})$
- Inverting $f(x_{b^*})$ is clearly hard even when given x_{1-b^*} and $f(x_{1-b^*})$
- An inverter can guess the forged bit b^*ahead of time w.p. $1/2$

Lamport's One-Time Scheme

Inverter $\mathcal{B}$:

Input: $y = f(x)$ for some $x \leftarrow \{0,1\}^n$.

1. Choose $b^* \leftarrow \{0,1\}$, and set $y_{b^*} = y$.
2. Sample $x_{1-b^*} \leftarrow \{0,1\}^n$ and set $y_{1-b^*} = f(x_{1-b^*})$.
3. Run $\mathcal{A}$ on input $vk = (y_0, y_1)$.
4. When $\mathcal{A}$ requests a signature on b:
   - If $b = b^*$, abort.
   - If $b = 1 - b^*$ output x_{1-b^*}.
5. If $\mathcal{A}$ output a forgery σ^* on b^*, output σ^*.

Independence!

$$\Pr[\mathcal{B}(f(x)) \in f^{-1}(f(x))] \geq \Pr[\mathrm{SigForge}_{\Pi,\mathcal{A}}(n) = 1 \ \wedge \mathcal{B} \text{ doesn't abort}]$$

$$= \Pr[\mathrm{SigForge}_{\Pi,\mathcal{A}}(n) = 1] \cdot \Pr[\mathcal{B} \text{ doesn't abort}]$$

$$= \Pr[\mathrm{SigForge}_{\Pi,\mathcal{A}}(n) = 1] \cdot \frac{1}{2}$$

Lamport's One-Time Scheme

Let f be a OWF. Define a signature scheme $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ for ℓ-bit messages as follow:

- $\mathrm{Gen}(1^n)$: For each $i \in [\ell]$ and $b \in \{0,1\}$ sample $x_{i,b} \leftarrow \{0,1\}^n$ and compute $y_{i,b} = f(x_{i,b})$. Output $sk = \{(x_{i,0}, x_{i,1})\}_{i\in[\ell]}$ and $vk = \{(y_{i,0}, y_{i,1})\}_{i\in[\ell]}$.
- $\mathrm{Sign}_{sk}(m = m_1 \cdots m_\ell)$: Output $\sigma = (x_{1,m_1}, \dots, x_{\ell,m_\ell})$.
- $\mathrm{Vrfy}_{vk}(m = m_1 \cdots m_\ell, \sigma = (x_1, \dots, x_\ell))$: If $f(x_i) = y_{i,m_i}$ for all $i \in [\ell]$ output 1, and otherwise output 0.

Theorem:
If f is a OWF then Π is a secure one-time signature scheme for ℓ-bit messages.

Proof idea:

- Suppose that $\mathcal{A}$ asks for a signature on m and then forges on $m^* \neq m$
- The inverter $\mathcal{B}$ needs to guess $i \in [\ell]$ s.t. $m_i^* \neq m_i$ as well as guess the bit m_i^*

One-Time Signatures -- Summary

Theorem (Lamport '79):
If OWFs exist then for any polynomial $\ell = \ell(n)$ there is a one-time signature scheme for signing ℓ-bit messages.

The following theorem is known as the "Hash-and-Sign" paradigm:

Theorem:
If CRHFs exist then there is a one-time signature scheme that can sign messages of arbitrary polynomial length.

Construction Outline

One-time signature scheme

↓

Stateful signature scheme

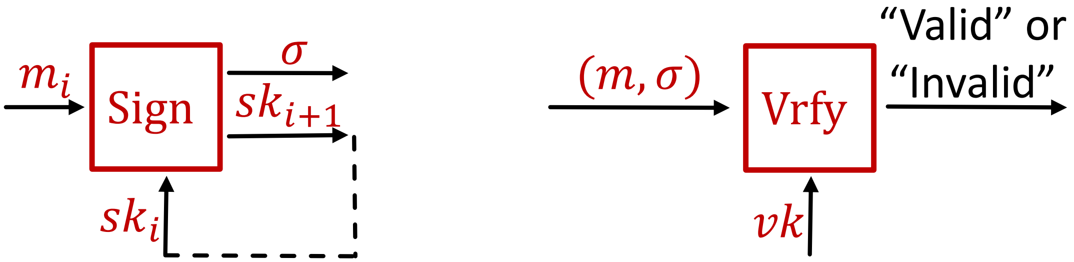
↓

Stateless signature scheme

Stateful Signature Schemes

Signer updates the signing key after each signature

- Initial state sk_1 produced by Gen: $(vk, sk_1) \leftarrow \mathrm{Gen}(1^n)$
- Signing the ith message updates sk_i to sk_{i+1}: $(\sigma, sk_{i+1}) \leftarrow \mathrm{Sign}_{sk_i}(m_i)$
- Verification requires only vk



Existential unforgeability against an adaptive chosen-message attack

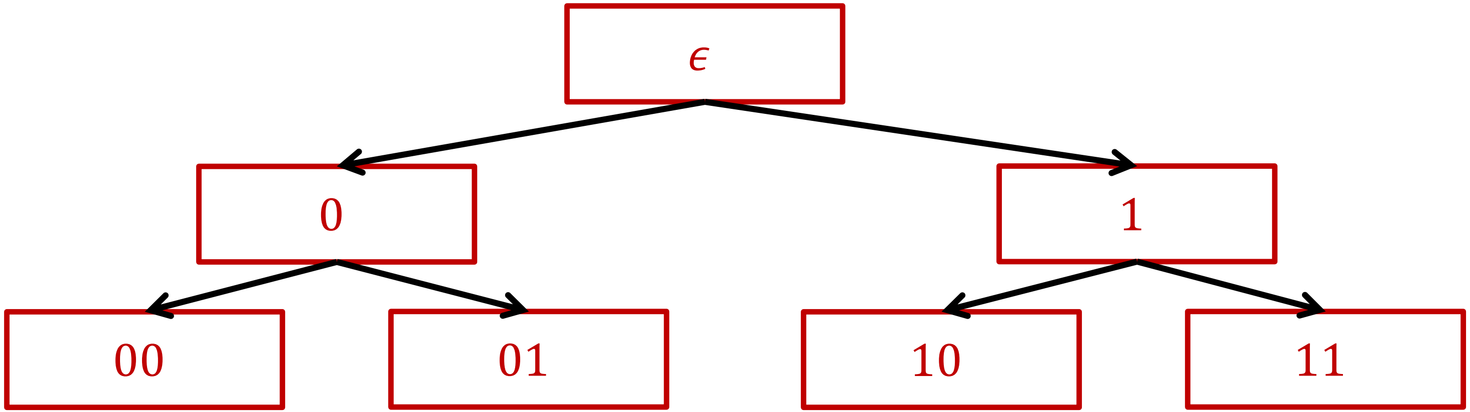
- $\mathcal{A}$ knows vk and can adaptively ask for signatures of messages of its choice
- The signing oracle maintains the internal state sk_i
- $\mathcal{A}$ tries to forge a signature on a new message

A Stateful Scheme

- Let $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ be a one-time signature scheme for signing "sufficiently long" messages
- For $m = m_1 \cdots m_n \in \{0,1\}^n$ we let $m|_i \stackrel{\text{def}}{=} m_1 \cdots m_i$ (and $m|_0 \stackrel{\text{def}}{=} \epsilon$)

Define $\boldsymbol{\Pi' = (\mathrm{Gen}', \mathrm{Sign}', \mathrm{Vrfy}')}$ for signing $\boldsymbol{n}$-bit messages as follows:

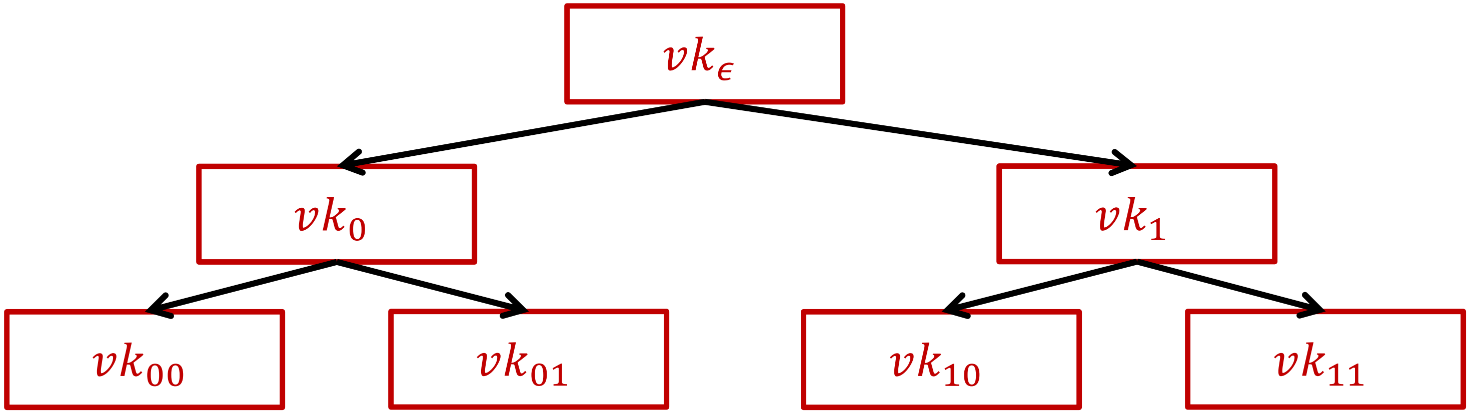
- The signer's state is binary tree with 2^n leaves
- Each node $w \in \{0,1\}^{<n}$ has a left child $w0$ and a right child $w1$
- The tree is of exponential size but is never fully constructed



A Stateful Scheme

Key generation:

- Each node $w \in \{0,1\}^{\leq n}$ is associated with $(vk_w, sk_w) \leftarrow \mathrm{Gen}(1^n)$
- Keys are generated and stored only when needed
- The state sk'_i consists of all keys and signatures that were generated so far
- $vk' = vk_\epsilon$ and $sk'_1 = sk_\epsilon$

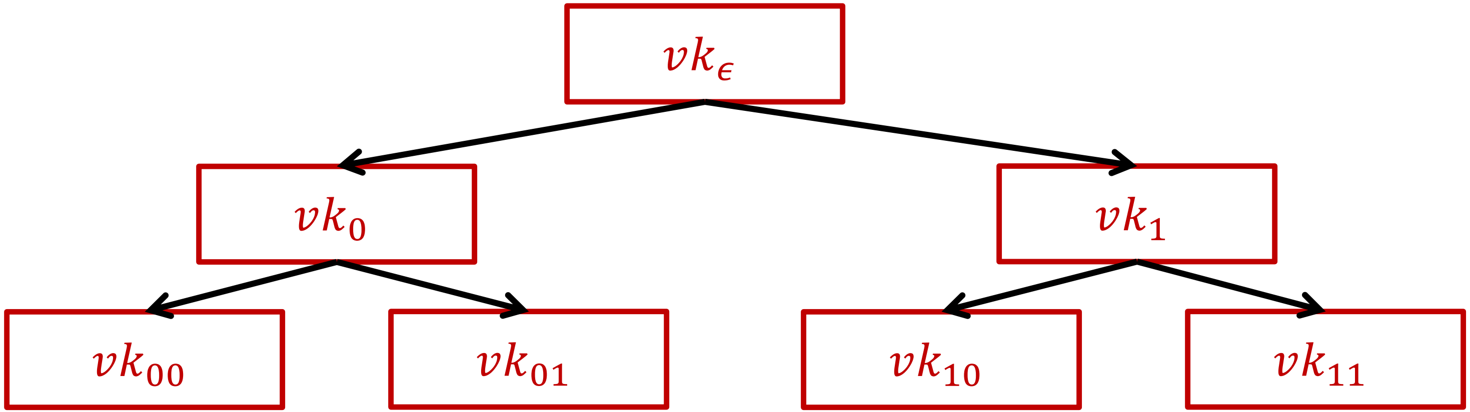


A Stateful Scheme

Signing a message $\boldsymbol{m \in \{0,1\}^n}$:

1. Generate a path from the root to the leaf labeled m: For each proper prefix w of m sample $(vk_{w0}, sk_{w0}), (vk_{w1}, sk_{w1}) \leftarrow \mathrm{Gen}(1^n)$
2. Certify the path: For each proper prefix w of m compute $\sigma_w = \mathrm{Sign}_{sk_w}(vk_{w0}, vk_{w1})$
3. Compute $\sigma_m = \mathrm{Sign}_{sk_m}(m)$

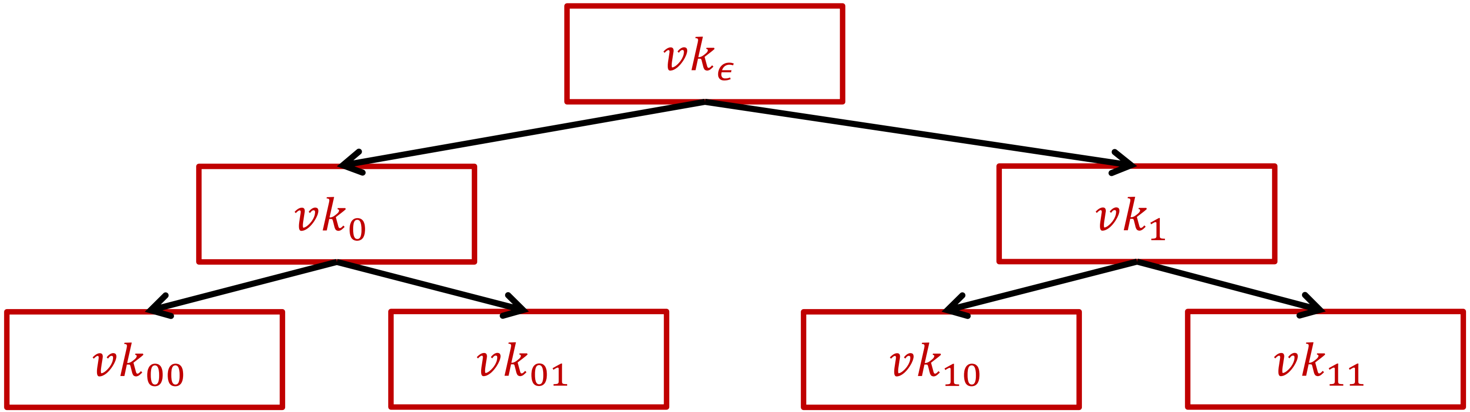
Values are generated in steps 1-3 only if these values are not already part of the current state



A Stateful Scheme

Signing a message $\boldsymbol{m} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$:

1. Generate a path from the root to the leaf labeled m: For each proper prefix w of m sample $(vk_{w0}, sk_{w0}), (vk_{w1}, sk_{w1}) \leftarrow \mathrm{Gen}(1^n)$
2. Certify the path: For each proper prefix w of m compute $\sigma_w = \mathrm{Sign}_{sk_w}(vk_{w0}, vk_{w1})$
3. Compute $\sigma_m = \mathrm{Sign}_{sk_m}(m)$
4. Store all generated keys and signatures as part of the updated state
5. Output the signature $\left(\left\{\sigma_{m|_i}, vk_{m|_i0}, vk_{m|_i1}\right\}_{i=0}^{n-1}, \sigma_m\right)$



A Stateful Scheme

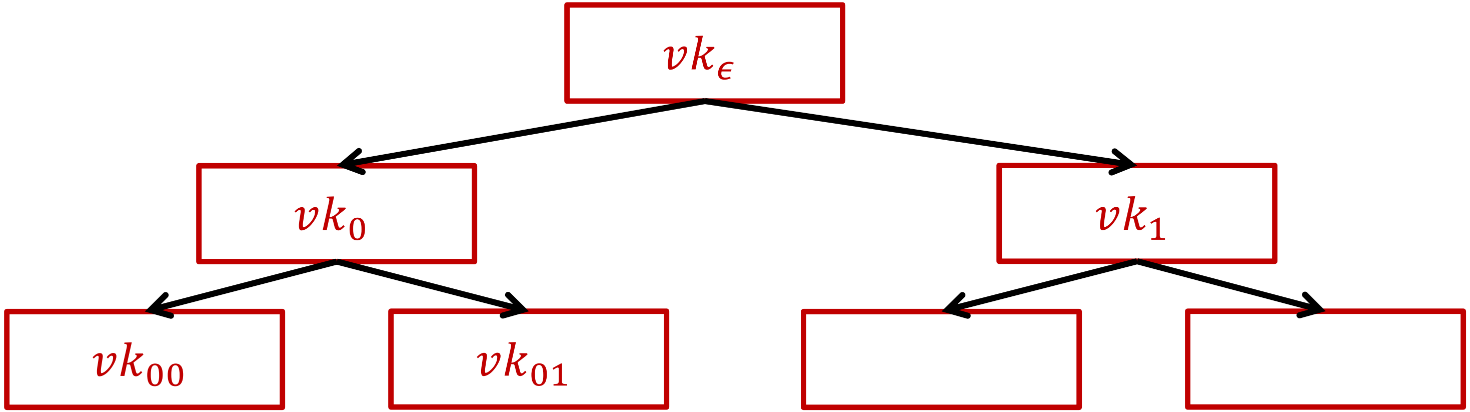
Example: A signature on $m = 01$ consists of $(\sigma_\epsilon, \sigma_0, \sigma_{01})$ where

$$\sigma_\epsilon = \mathrm{Sign}_{sk_\epsilon}(vk_0, vk_1)$$
$$\sigma_0 = \mathrm{Sign}_{sk_0}(vk_{00}, vk_{01})$$

(Certifying the path)

$$\sigma_{01} = \mathrm{Sign}_{sk_{01}}(01)$$

(Signing the message)

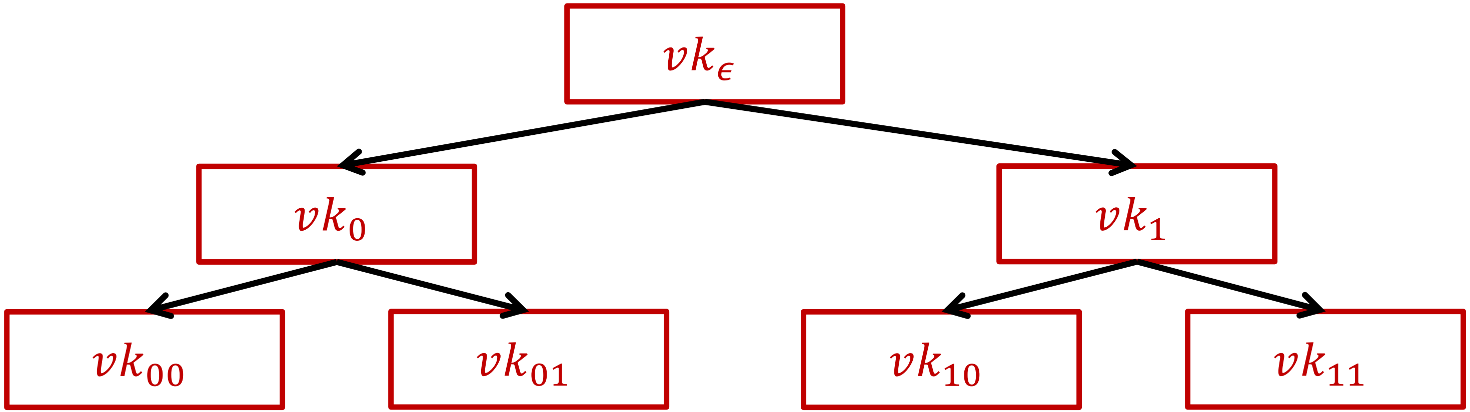


A Stateful Scheme

Verifying a signature $\left(\left\{\boldsymbol{\sigma}_{\boldsymbol{m}|_i}, \boldsymbol{vk}_{\boldsymbol{m}|_i 0}, \boldsymbol{vk}_{\boldsymbol{m}|_i 1}\right\}_{\boldsymbol{i}=0}^{\boldsymbol{n}-1}, \boldsymbol{\sigma}_{\boldsymbol{m}}\right)$ **on** $\boldsymbol{m} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$**:**

Output 1 if and only if both:

1. $\mathrm{Vrfy}_{vk_{m|_i}}\left(\left(vk_{m|_i 0}, vk_{m|_i 1}\right), \sigma_{m|_i}\right) = 1$ for every $i \in \{0, \dots, n-1\}$
2. $\mathrm{Vrfy}_{vk_m}(m, \sigma_m) = 1$



A Stateful Scheme

Theorem:
If Π is a one-time signature scheme, then Π' is existentially unforgeable against a chosen-message attacks.

Note:
Π needs to allow signing "sufficiently long" messages (two verification keys of Π)

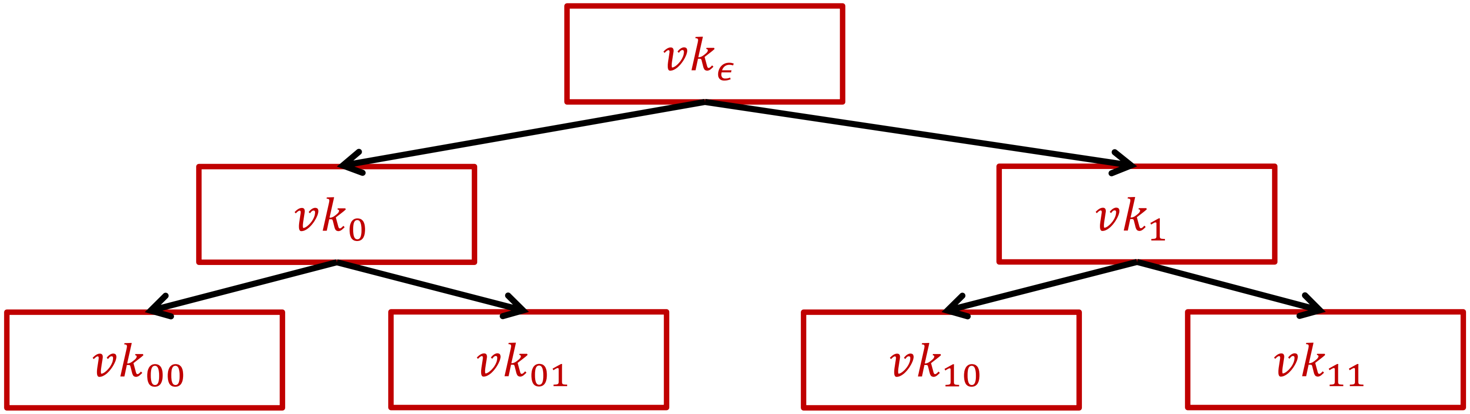
- Can be constructed from CRHFs by applying the hash-and-sign paradigm to Lamport's scheme
- In fact, can be constructed assuming OWFs instead of CRHFs (but this is outside the scope of this course)

A Stateful Scheme

Theorem:
If Π is a one-time signature scheme, then Π' is existentially unforgeable against a chosen-message attacks.

Proof idea #1:
Each sk_w is used to sign exactly one "message"

- If w is an internal node then sk_w is used to sign (vk_{w0}, vk_{w1})
- If w is a leaf then sk_w is used to sign w



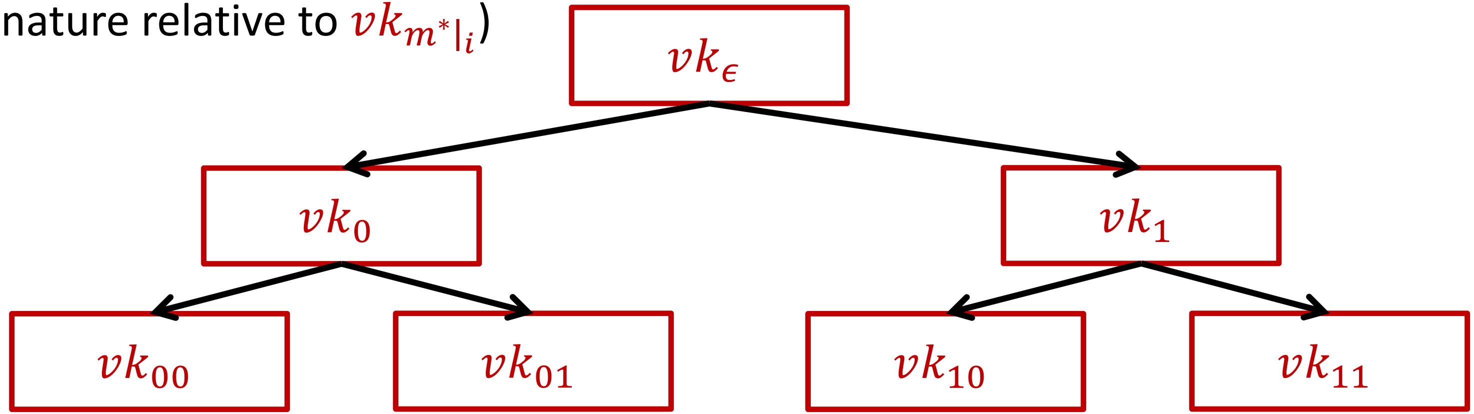
A Stateful Scheme

Proof idea #2:

Suppose that $\mathcal{A}$ asks forges a signature $\left(\left\{\sigma^*_{m^*|_i}, vk^*_{m^*|_i 0}, vk^*_{m^*|_i 1}\right\}_{i=0}^{n-1}, \sigma^*_{m^*}\right)$ on m^*.

Two possible cases:

- The full path to the leaf m^* already existed and $\mathcal{A}$ used the same path $\Rightarrow$ $\mathcal{A}$ must have forged a signature relative to vk_{m^*} (and did not receive any signature relative to vk_{m^*})
- The full path to the leaf m^* didn't exist or $\mathcal{A}$ used a different path $\Rightarrow$ $\mathcal{A}$ must have forged a signature relative to $vk_{m^*|_i}$ for $i \in \{0, \ldots, n-1\}$ (and received exactly one signature relative to $vk_{m^*|_i}$)



Construction Outline

One-time signature scheme

↓

Stateful signature scheme

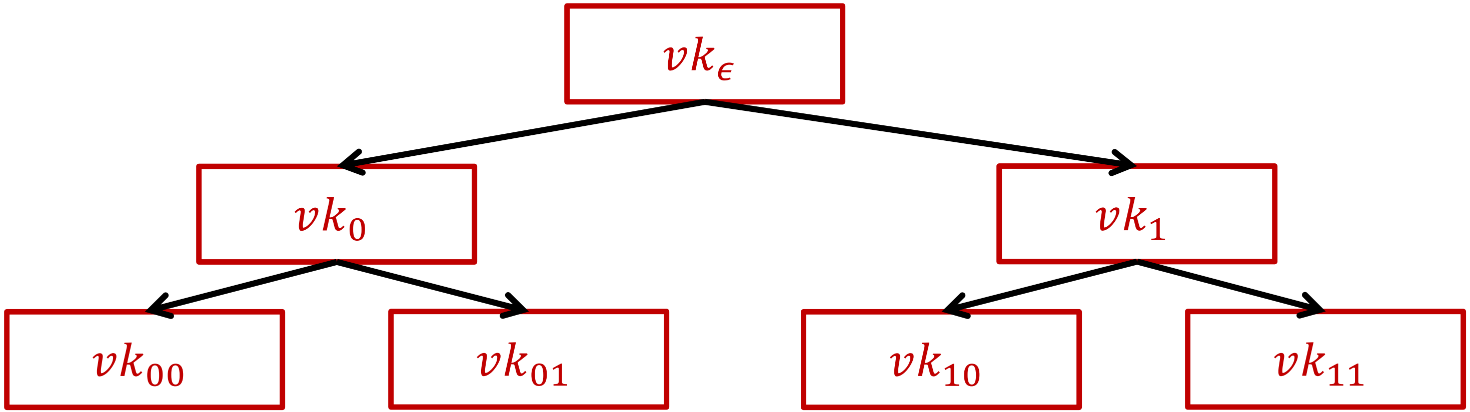
↓

Stateless signature scheme

A Stateless Scheme

De-randomize the stateful scheme Π' to a stateless scheme Π'':

- The signer's secret key sk is a seed for a PRF $F_{sk}(\cdot)$
- $(r_w, r_w') \stackrel{\text{def}}{=} F_{sk}(w)$ is used as the randomness needed for each node $w \in \{0,1\}^{\leq n}$:
  - If $w \in \{0,1\}^{<n}$ then r_w is used for sampling (vk_w, sk_w) and r_w' is used for signing (vk_{w0}, vk_{w1})
  - If $w \in \{0,1\}^n$ then r_w is used for sampling (vk_w, sk_w) and r_w' is used for signing w



A Stateless Scheme

De-randomize the stateful scheme Π' to a stateless scheme Π'':

- The signer's secret key sk is a seed for a PRF $F_{sk}(\cdot)$
- $(r_w, r'_w) \stackrel{\text{def}}{=} F_{sk}(w)$ is used as the randomness needed for each node $w \in \{0,1\}^{\leq n}$
  - If $w \in \{0,1\}^{<n}$ then r_w is used for sampling (vk_w, sk_w) and r'_w is used for signing (vk_{w0}, vk_{w1})
  - If $w \in \{0,1\}^{n}$ then r_w is used for sampling (vk_w, sk_w) and r'_w is used for signing w

Theorem:
If Π is a one-time signature scheme and F is a PRF, then Π'' is existentially unforgeable against a chosen-message attacks.

Proof idea:
Any adversary $\mathcal{A}$ against Π'' can be used either as an adversary against the stateful scheme Π', or as a distinguisher against the PRF F

A Stateless Scheme

Theorem:
If Π is a one-time signature scheme and F is a PRF, then Π'' is existentially unforgeable against a chosen-message attacks.

$$\Pr\left[\mathrm{SigForge}_{\Pi'',\mathcal{A}}(n)=1\right] \le \left|\Pr\left[\mathrm{SigForge}_{\Pi'',\mathcal{A}}(n)=1\right]-\Pr\left[\mathrm{SigForge}_{\Pi',\mathcal{A}}(n)=1\right]\right|$$

$$+\Pr\left[\mathrm{SigForge}_{\Pi',\mathcal{A}}(n)=1\right]$$

$$=\left|\Pr\left[\mathcal{D}^{F_{sk}(\cdot)}(1^n)=1\right]-\Pr\left[\mathcal{D}^{f(\cdot)}(1^n)=1\right]\right|$$

$$+\Pr\left[\mathrm{SigForge}_{\Pi',\mathcal{A}}(n)=1\right]$$

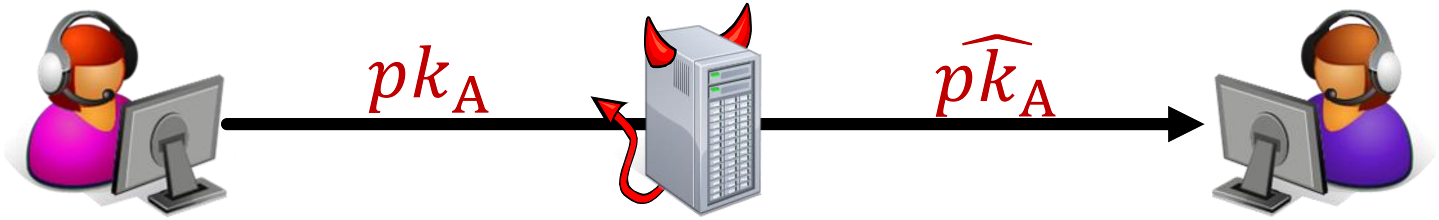
Outline

- **Digital signatures**
- **Constructions**
  - One-time signatures
  - Stateful signatures
  - Stateless signatures
- $\boxed{\textbf{Certificates and public-key infrastructures}}$
- **User-server identification**

Certificates and PKI

Public-key cryptography is great, but how to distribute the public keys?

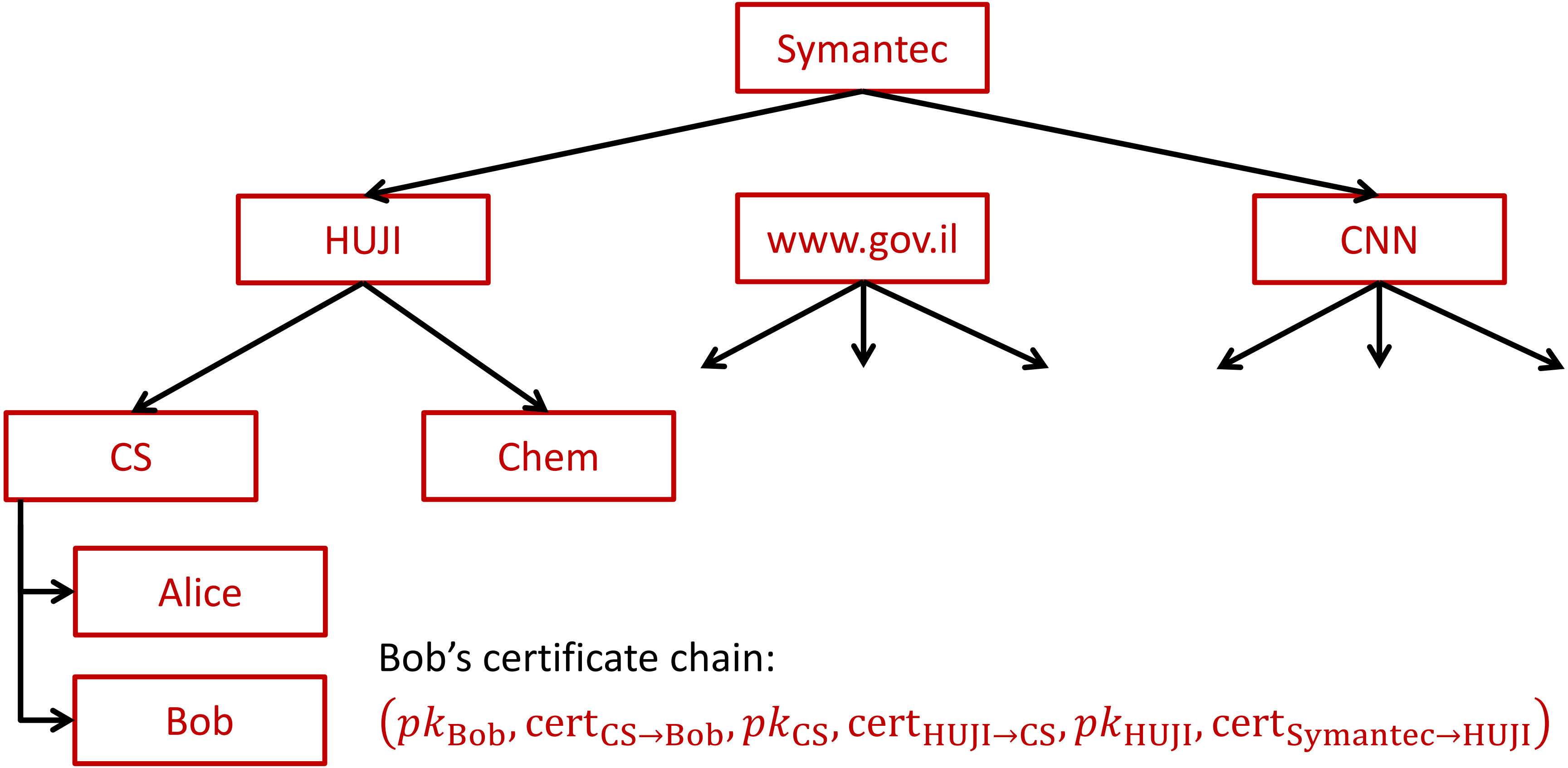
- Keys must be authenticated for avoiding man-in-the-middle attacks



Solution: Certification Authorities (CAs)

- Certificate: A signature binding an identity to a public key
- Assume that we already trust the CA's verification key vk_{CA} (e.g., vk_{CA} is hard-wired into the source code of my browser)
- The CA provides Alice with $\mathrm{cert}_{\mathrm{CA}\to\mathrm{A}} \stackrel{\mathrm{def}}{=} \mathrm{Sign}_{sk_{\mathrm{CA}}}(\text{"Alice's key is } pk_{\mathrm{A}}\text{"})$
- Alice sends to Bob both pk_{A} and $\mathrm{cert}_{\mathrm{CA}\to\mathrm{A}}$

Delegation of Certificates



Bob's certificate chain:

$\left(pk_{\mathrm{Bob}}, \mathrm{cert}_{\mathrm{CS}\to\mathrm{Bob}}, pk_{\mathrm{CS}}, \mathrm{cert}_{\mathrm{HUJI}\to\mathrm{CS}}, pk_{\mathrm{HUJI}}, \mathrm{cert}_{\mathrm{Symantec}\to\mathrm{HUJI}}\right)$

Invalidating Certificates

Certificates should not be valid indefinitely

- An employee may leave a company
- A secret key may get stolen
- …

Approach 1: Expiration

- Each certificate includes an expiration date
- $\mathrm{cert}_{\mathrm{CA}\to\mathrm{A}} \stackrel{\mathrm{def}}{=} \mathrm{Sign}_{sk_{\mathrm{CA}}}(\text{"Alice's key is } pk_{\mathrm{A}}\text{"}, 31/12/2014)$

Approach 2: Revocation

- Each certificate includes a unique serial number
- The CA publishes (a signed) list of revoked certificates
- $\mathrm{cert}_{\mathrm{CA}\to\mathrm{A}} \stackrel{\mathrm{def}}{=} \mathrm{Sign}_{sk_{\mathrm{CA}}}(\text{"Alice's key is } pk_{A}\text{"}, \text{ serial number})$

Outline

- **Digital signatures**
- **Constructions**
  - One-time signatures
  - Stateful signatures
  - Stateless signatures
- **Certificates and public-key infrastructures**
- **User-server identification**

User-Server Identification

A trivial password-based identification protocol

- The user holds a password pwd, the server knows $y = f(pwd)$ for some function f
- The user identifies by sending pwd in the clear…

User
pwd

What is your password? $\longleftarrow$

pwd $\longrightarrow$

Server
$y = f(pwd)$

User-Server Identification

A slightly better solution using a signature scheme

- The user holds a signing key sk, the server knows the verification key vk
- The user identifies by signing a randomly chosen message

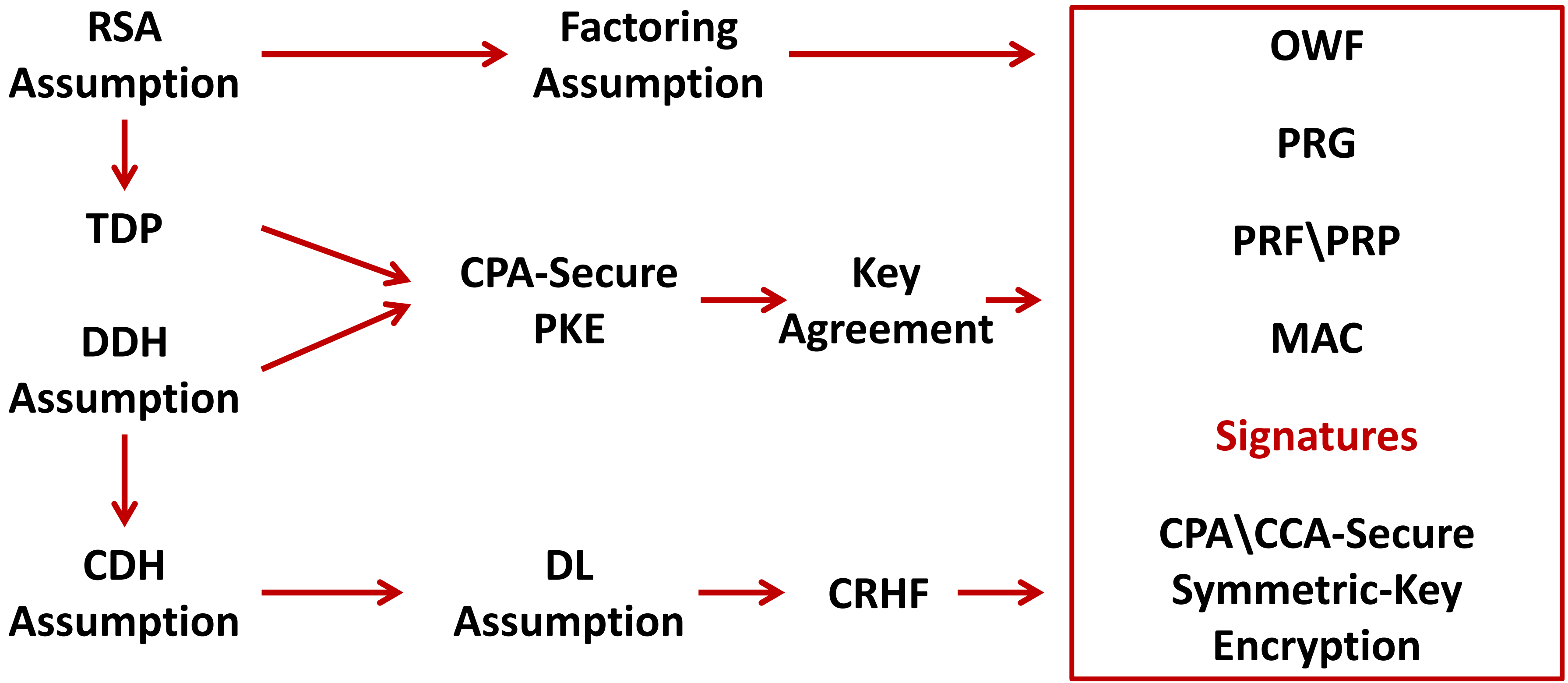
$$r \leftarrow \{0,1\}^n$$

$$\mathrm{Sign}_{sk}(r, \text{"UserName"}, \text{"ServerName"})$$

User
sk

Server
vk

The World of Crypto Primitives



Recommended Reading

- J. Katz and Y. Lindell. **Introduction to Modern Cryptography**.
  Chapter 12 (Digital Signature Schemes): 12.0-12.3, 12.6-12.7

Problem set 5 is available on-line