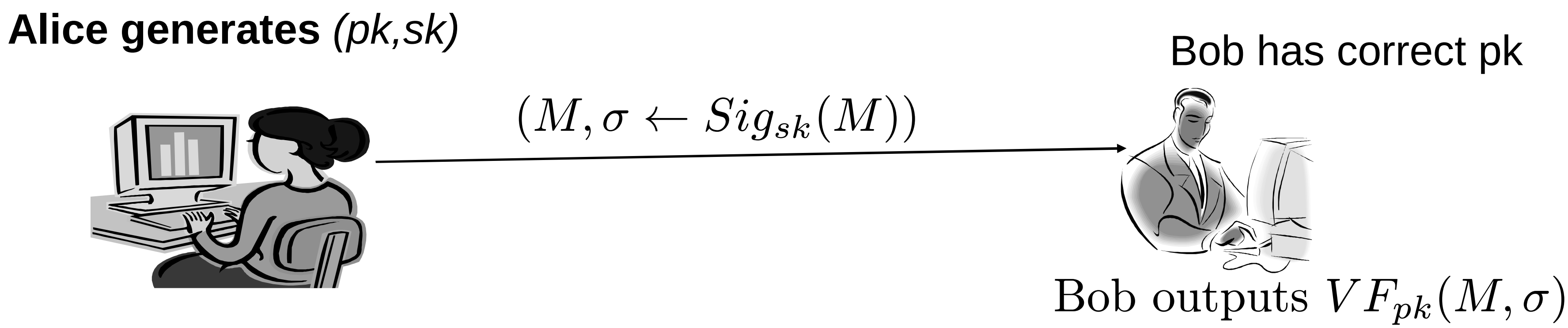
Digital Signatures

Murat Kantarcioglu

Digital Signatures

- Define a digital signature scheme $DS = (\mathcal{K}, Sign, VF)$
- Key generation: $(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}$
- Signing a message: $\sigma \stackrel{\$}{\leftarrow} Sign_{sk}(M)$
- Signature Verification $d \stackrel{\$}{\leftarrow} VF_{pk}(M, \sigma)$
  - $d = 1$ if σ is valid for
    for given message under (pk, sk) pair
  - else $d = 0$

Digital Signature Assumptions



- ★ Bob assumed to have correct pk
- ★ Sender (Alice) has the private key
- ★ Sig could be randomized and /or stateful
- ★ We will mainly focus on deterministic Sig algorithms
  - ▶ Unlike PKE algorithms

Defining Security

Definition 9.2 Let $\mathcal{DS} = (\mathcal{K}, \mathrm{Sign}, \mathrm{VF})$ be a digital signature scheme, and let A be an algorithm that has access to an oracle and returns a pair of strings. We consider the following experiment:

Experiment $\mathbf{Exp}_{\mathcal{DS}}^{\text{uf-cma}}(A)$

$(pk, sk) \xleftarrow{\$} \mathcal{K}$

$(M, \sigma) \leftarrow A^{\mathrm{Sign}_{sk}(\cdot)}(pk)$

If the following are true return 1 else return 0:

- $\mathrm{VF}_{pk}(M, \sigma) = 1$
- $M \in \mathsf{Messages}(pk)$
- M was not a query of A to its oracle

The *uf-cma-advantage* of A is defined as

$$\mathbf{Adv}_{\mathcal{DS}}^{\text{uf-cma}}(A) = \Pr\left[\mathbf{Exp}_{\mathcal{DS}}^{\text{uf-cma}}(A) = 1\right].$$

pk

(M, σ) for any M

A:

M_1, σ_1

M_2, σ_2

$\vdots$

M_q, σ_q

$M \in \{M_1, \ldots, M_q\}$

RSA based Signatures

- ★ $((N, e), (N, p, q, d)) \leftarrow (K)$ where $e.d = 1 \bmod \phi(N)$, $N = pq$

- ★ Signature Generation
  - ▶ Algorithm $Sign_{N,p,q,d}(M)$
  - ▶ if $M \in Z_N^*$ return $\perp$
  - ▶ return $M^d \bmod N$

- ★ Verification
  - ▶ Algorithm $VF_{N,e}(M, \sigma)$
  - ▶ if $M \notin Z_N^* \vee \sigma \notin Z_N^*$ return 0
  - ▶ if $M = \sigma^e \bmod N$ return 1 else 0

- ★ Direct RSA signature generation is not secure

Possible Attacks

★ Forger F_1

- Forger $F_1^{Sign_{N,p,q,d}()}(N, e)$
- return $(1, 1)$

$1^d \bmod N = 1$

$(1, 1)$

$Adv_{DS}^{uf\text{-}cma}(F_1) = 1$

★ Forger F_2

- Forger $F_2^{Sign_{N,p,q,d}()}(N, e)$
- $\sigma \leftarrow Z_N^*$, $M \leftarrow \sigma^e \bmod N$
- return (M, σ)

$Adv_{DS}^{uf\text{-}cma}(F_2) = 1$

★ Forger F_3

- Forger $F_3^{Sign_{N,p,q,d}()}(N, e)$
- $\mathrm{M}_1 \leftarrow Z_N^* - \{1, M\}$, $M_2 \leftarrow MM_1^{-1} \bmod N$
- $\sigma_1 \leftarrow Sign_{N,p,q,d}(M_1), \sigma_2 \leftarrow Sign_{N,p,q,d}(M_2)$
- return $(M, \sigma_1\sigma_2 \bmod N)$

All attacks have advantage one

Hash-then-invert paradigm

- ★ Goal: RSA based scheme that
  - ▶ is provably secure
  - ▶ has Flexible message space

- ★ Idea Hash the message first given $H_N : \{0,1\}^* \mapsto Z_N^*$

- ★ Signature Generation
  - ▶ Algorithm $Sign_{N,p,q,d}(M)$
  - ▶ $y \leftarrow H_N(M)$
  - ▶ return $y^d \bmod N$

- ★ Verification
  - ▶ Algorithm $\ddot{V}F_{N,e}(M,\sigma)$
  - ▶ $y \leftarrow H_N(M)$
  - ▶ if $y = \sigma^e \bmod N$ return 1 else 0

Hash then Invert Paradigm

- ★ Previous Forgers described do not work well for Hash-then-Invert
  - $H_N(1) \neq 1$ with high probability (w.h.p)
  - $\sigma^e \bmod N \neq H_N(M)$ w.h.p
  - $H_N(M_1).H_N(M_2) \neq H_N(M)$ w.h.p

- ★ Not secure if it is easy to find $M_1 \neq M$ such that $H_N(M_1) = H_N(M_2)$

- ★ What are the assumptions needed to make Hash then Invert Paradigm Secure??