Digital Signatures

One-Way Functions.

Def: an OWF is a fun $F: X \to Y$ s.t.

1. $\exists$ "eff." alg to eval. F
2. $\forall$ "eff." alg A:

$$\Pr[F[A(F(x)] = F[x]] \text{ is negligible}$$

where $x \xleftarrow{R} X$ → inversion/preimage

given $y = f(x)$, hard to find preimage x of y

Ex.

1. General OWF: Let (E, D) be block cipher

$$F^E(k) = [E(k,0), E(k,1), \ldots E(k,10)]$$

No special props, bad for key exch

2. G grp of order q w/gen $g \in G$

w/ gen $g \in G$

$$F^{Dlog}(x) = g^x \in G$$

$$F^{Dlog} : \mathbb{Z}_q \to G$$

inversion: Dlog in G base g

props: $F(x)F(y) = F(x+y)$

$$F(x)^d = F(dx) \qquad d \in \mathbb{Z}$$

⇒ DH key exch and ElGamal

3. RSA $n = pq$, $e \in \mathbb{Z}^*_{\phi(n)}$

$$F^{RSA}(x) = x^e \text{ in } \mathbb{Z}_n$$

inversion: RSA assumption

$$F^{RSA} : \mathbb{Z}_n \to \mathbb{Z}_n$$

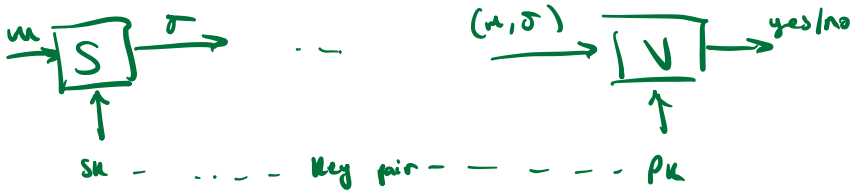
props: 1. $F^{RSA}(x) \cdot F^{RSA}(y) = F^{RSA}(xy)$

2. Trapdoor $d = e^{-1} \bmod \phi(n)$

→ RSA enc, signatures

Digital signatures

My dig. sig. on m is a fn on m



Def: A sig. scheme is a tuple of algs (Gen, S, V)

- $Gen() \to PK, SK$
- $S(SK, m) \to \sigma$
- $V(PK, m, \sigma) \to$ yes/no (deterministic)

s.t. if $(PK, SK) \leftarrow Gen()$ then

$$\forall m \in M: V(PK, m, S(SK, m)) = \text{"yes"}$$

note: signer signs m once $\to \sigma$ $\leftarrow$ one person

anyone w/ PK, m can verify σ $\leftarrow$ many people

Security

Attacker power: chosen msg attack

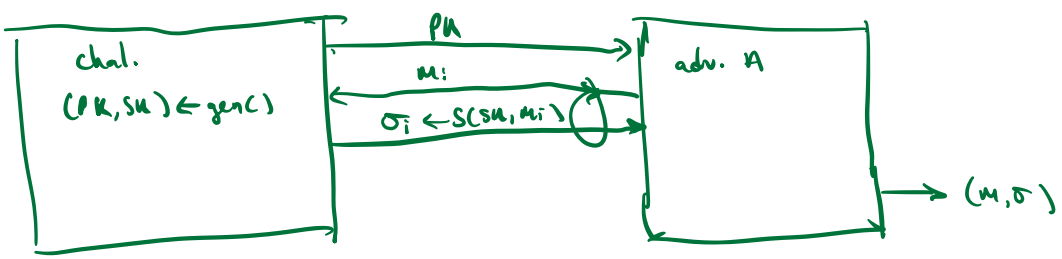
$\to$ for $m_1, \dots m_\ell \in M$: attacker given $\sigma_i \leftarrow S(SK, m_i)$

Attacker goal: existential forgery

$\to$ produce some new valid pair (m, σ)

s.t. $m \notin \{m_1, \dots m_\ell\}$

For sig scheme (Gen, S, V) and adv A:



Adv. wins if $V(PK, m, \sigma) =$ "yes" and $m \notin \{m_1, \dots m_\ell\}$

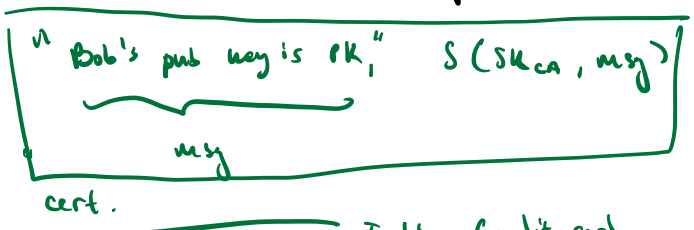
Def. Sig scheme (Gen, S, V) secure if $\forall$ eff. adv A:

$\Pr[A \text{ wins game}] \leq$ negligible

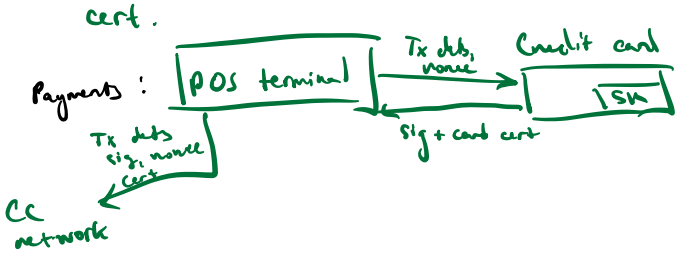
Applications

1. Software updates: Android ships w/ built-in PK
   every update is signed by SK
   phones will ignore updates not properly signed

2. Certificates: CA creates signed message



3. Payments:



Overall: Approaches to data integrity

1. CRH: need read-only public space for hashes
   many verifiers

2. Dig. sig: one signer (SK), many verifiers (PK)
   verifier needs correct PK.

3. MAC: one signer, one verifier
   (K) (K)

Extending the domain of a signature scheme

Let $Sig = (Gen, S, V)$ be a scheme for short msgs $M = \{0,1\}^{256}$

Let $H: M^{big} \to M$ be a CRH (SHA256)

Def: $S^{big}(SK, m) = S(SK, H(m))$

$V^{big}(PK, m, \sigma) = V(PK, H(m), \sigma)$

Thm: If Sig is secure, H is CRH, then SigBig is secure.

⇒ Suffices to build sig. scheme for short msgs, then use SHA256 to expand.

Primitives that imply signature schemes.

1. Sig. scheme from general OWF: $f : X \to Y$

   ex. Lamport-Merkle (XMSS)

   Problem: sigs are long

   → stateless signer: 30kB

   → stateful signer: 4kB

   Suitable for software updates.

   Quantum-resistant.

2. Dlog: $F(x) = g^{(x)}$ in G

   ex. ECDSA, Schnorr, BLS

   Sig. size: 48 or 64 bytes

   PK size: 32 or 48 bytes

3. Trapdoor permutation $f : X \to X$ (RSA)

Sigs from Trapdoor Permutation

→ (Gen, F, F^{-1}):

$F(pk, \cdot) : X \to X$

$F^{-1}(sk, \cdot) : X \to X$

$\forall x \in X : F^{-1}(sk, F(pk, x)) = x$

→ $H : M \to X$ hash func

Sig scheme:

Gen: same as TDP

$S(sk, m \in M)$: output $F^{-1}(sk, H(m)) \to \sigma$

$V(pk, m \in M, \sigma \in X)$: $F(pk, H(m)) = \sigma \Rightarrow$ output yes, else no

Thm: (Gen, S, V) is a secure sig. scheme
assuming $(Gen, \mathcal{P}, \mathcal{P}^{-1})$ is a secure TDP
and H is a "random oracle"

RSA-FDH (full domain hash)

Gen: 1. choose $n = p \cdot q$
e, d s.t. $ed = 1 \pmod{\varphi(n)}$ large sig, 2-3 KB

2. output $PK = (n, e)$, $SK = (n, d)$

Ingredient: $H: M \to \mathbb{Z}_n$ (FDH)

$S(SK, m)$: $\sigma \leftarrow [H(m)^d \in \mathbb{Z}_n]$

$V(PK, m, \sigma)$: accept iff $[\sigma^e \text{ in } \mathbb{Z}_n] = H(m)$

Problem: range of H depends on PK
note: $e=3 \Rightarrow$ superfast sig. verify

Why hash the msg? (why not $S(SK, m) := [m^d \text{ in } \mathbb{Z}_n]$?)
insecure!
attack 1: forging given PK
step 1: choose $\sigma \in \mathbb{Z}_n$
step 2: compute $m = \sigma^e$ in $\mathbb{Z}_n$
output m, σ as forgery
Then $V(PK, m, \sigma) =$ yes b/c
$\sigma^e = m$ in $\mathbb{Z}_n$!
$\Rightarrow$ existential forgery

attack 2: (on $\sigma \leftarrow m^d$ in $\mathbb{Z}_n$)
Adv. has $PK = (n, e)$ does:
1. Choose $r \xleftarrow{R} \mathbb{Z}_n$
compute $\hat{m} \leftarrow r^e \cdot m$ in $\mathbb{Z}_n$
2. Request sig on $\hat{m} \in \mathbb{Z}_n$
get back $\hat{\sigma} \in \mathbb{Z}_n$ s.t. $\hat{\sigma}^e = \hat{m}$
3. Let $\sigma \leftarrow \hat{\sigma} / r$ in $\mathbb{Z}_n$

Claim: σ is a valid sig. for m

Proof: $\sigma^e = (\hat{\sigma} \cdot r)^e = \hat{m} \cdot r^e = m$

We forged sig. on m by asking for sig on $\hat{m}$

$\Rightarrow$ existential forgery!

$\Rightarrow$ blind sig. Chaum applications (anonymous cash, e-voting)

RSA in practice: PKCS1 v. 1.5

uses a hash for PKCS1: $M \to \mathbb{Z}_n$

$PKCS1(m) := [\ 0x1 \| FF \| FF \| \ldots \| 00 \| SHA256(m)\ ]$

(0x1 with)

RSA mod, 2048 bits

So: $\sigma = [\ PKCS1(m)^d \text{ in } \mathbb{Z}_n\ ]$

Problem: not FDH (full domain hash)

$\Rightarrow$ no sec. analysis based on RSA assumption