Digital signatures

Recap: public key encryption (Gen, E, D):

(1) ElGamal: security based on CDH & variants

(2) TDF such as RSA: secure when TDF is one-way

The big picture: one way functions (OWF)

Def: a <u>one-way</u> function $F: X \to Y$ satisfies

(1) there is an "eff" alg. to compute F

(2) $\forall$ "eff" algs. $\mathcal{A}$: $\Pr[F(\mathcal{A}(F(x))) = F(x)] < \text{"neg"}$

where $x \xleftarrow{R} X$

$\Rightarrow$ given $y = F(x)$ it is hard to find <u>any</u> pre-image of y.

one-way functions so far

(1) General owf: let (E,D) be a block cipher

$$\boxed{F^E(k) = (E(k,0),\ E(k,1),\ \ldots,\ E(k,n))}$$

No special properties. Bad for key exch.

(2) G finite cyclic group of order q with gen. $g \in G$.

$$\boxed{F^{Dlog}(x) = g^x} \qquad F^{Dlog} : \mathbb{Z}_q \to G$$

inversion $\equiv$ DLOG in (G, g)

properties: $F(x+y) = F(x) \cdot F(y)$

$F(x)^y = F(x \cdot y)$

$\Rightarrow$ DH exch. & ElGamal enc.

(3) RSA: $n = p \cdot q$, $e \in \mathbb{Z}^*_{\varphi(n)}$

$$\boxed{F^{RSA}(x) = x^e} \qquad F^{RSA} : \mathbb{Z}_n \to \mathbb{Z}_n$$

inversion $\equiv$ RSA assumption

properties: (1) $F^{RSA}(x) \cdot F^{RSA}(y) = F^{RSA}(x \cdot y)$

(2) Trapdoor: $d = e^{-1} \pmod{\varphi(n)}$

$\Rightarrow$ RSA enc. & RSA dig. sigs.

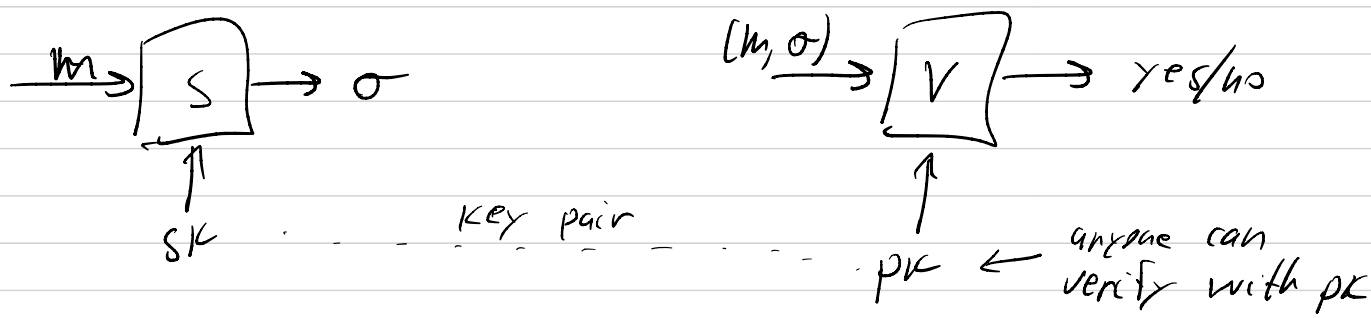
Digital signatures

Physical signature (check, contract, transcript) grades:

binds doc. to author.

doesn't work in digital world: easy to copy sig.

Instead: dig sig. is a function of document being signed.



Def: a <u>signature scheme</u> is a triple of algs. (Gen, S, V)

- $\text{Gen}() \to (pk, sk)$ key pair
- $S(sk, m \in \mathcal{M}) \to \sigma$
- $V(pk, m, \sigma) \to$ 'accept'/'reject' (det. alg.)

s.t. if $(pk, sk) \leftarrow \text{Gen}()$ then

$$\forall m \in \mathcal{M}: V(pk, m, S(sk, m)) = \text{'accept'}$$

<u>note</u>: $\underbrace{\text{sign once with } sk}_{\text{one signer}}$, anyone $\underbrace{\text{can verify with } pk}_{\text{many people}}$.

<u>security</u>:

<u>attacker's power</u>: chosen msg attack (CMA)

For $m_1, \ldots, m_Q \in \mathcal{M}$ attacker is given $\sigma_i \leftarrow S(sk, m_i)$

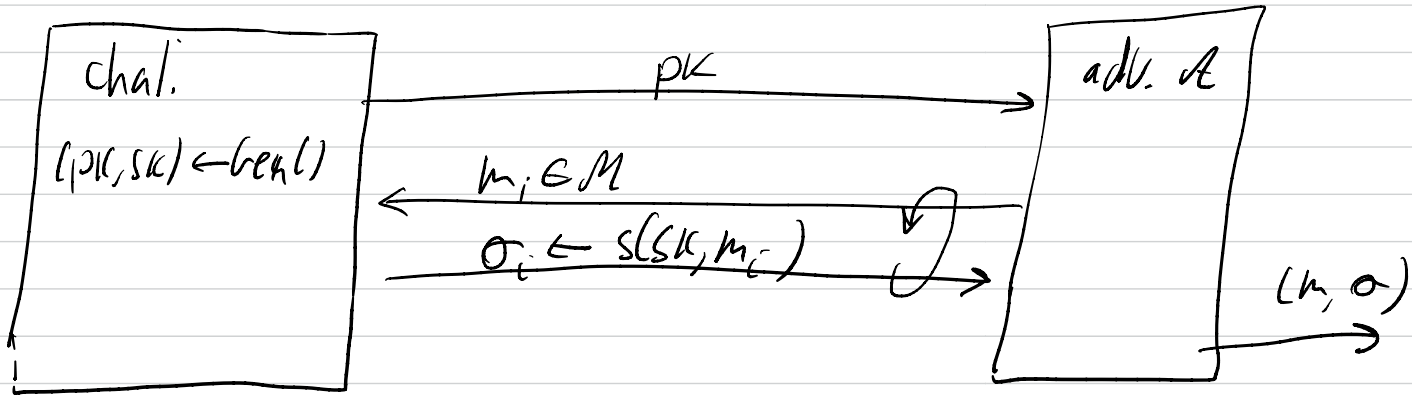
<u>attacker's goal</u>: existential forgery (EUF)

produce some <u>new</u> valid pair (m, σ)

s.t. $m \notin \{m_1, \ldots, m_Q\}$

Security: attacker cannot produce an ex. forgery using CMA

For a sig. scheme $Sig = (Gen, S, V)$ and adv. $\mathcal{A}$:



adv. win if $V(pk, m, \sigma) = \text{'accept'}$ & $m \notin \{m_1, m_2, \ldots\}$

<u>Def:</u> Sig is secure if $\forall$ eff. $\mathcal{A}$ $\Pr(\mathcal{A} \text{ wins}) < \text{"negl."}$

<u>many applications:</u>

(1) Android ships with built in pk.

every software update is signed by Google.

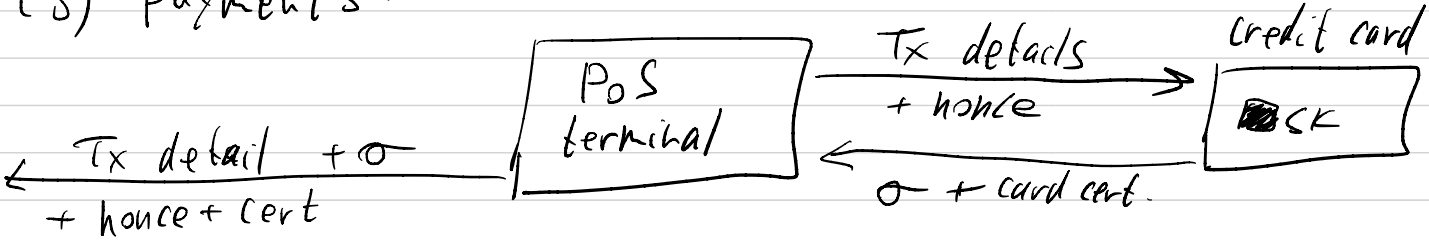
every device uses pk to verify sig. on update.

(2) certificates: CA creates a msg that says

$$\left[\text{"Bob's public key is pk"}, \quad S(sk_{CA}, msg)\right]$$

anyone who has pk_{CA} can verify cert.

(3) payments:



nonce prevents replay by merchant.

Three approaches to data integrity

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

(2) Dig. Sig.: one signer (SK) many verifiers
verifier need correct PK.

(3) MAC: one signer, one verifier.

Extending the domain of sig. scheme

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

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

<u>Def</u>: $Sig^{big} = (Gen, S^{big}, V^{big})$ for $M \in \mathcal{M}^{big}$

$$\begin{cases} S^{big}(SK, M) = S(SK, H(M)) \\ V^{big}(PK, m, \sigma) = V(PK, H(M), \sigma) \end{cases}$$

<u>Thm</u>: if Sig is secure and H is a CRH
then Sig^{big} is secure.

$\Rightarrow$ suffice to build Sig of short 256-bit msgs.

Primitives that imply signatures

(1) Sig scheme from a general OWF $F: X \to Y$??

Answer: yes! ex: Lamport-Merkle, XMSS

Problem: signatures are long:

stateless signer: ≥ 40 KB
stateful signer: ≥ 4 KB

} suitable for software update

quantum resistant

(2) DLOG: $F(x) = g^x \in G$??

Answer: yes! ex: ECDSA, Schnorr, BLS

sig size = 48 or 64 bytes (short)

will see in last lecture.

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

Answer: yes! very easy

sigs. from TDP (TDF not sufficient)

- $(\mathrm{Gen}, F, F^{-1})$ a TDP $\quad F(pk, \cdot) : X \to X$

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

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

- $H : \mathcal{M} \to X$ hash func.

The sig scheme (Gen, S, V) : (Gen is from TDP)

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

$V(pk, m, \sigma)$: accept iff $F(pk, \sigma) = H(m)$

intuition: only signer can invert F at $H(m)$.

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

RSA-FDH:

Gen: choose (1) $n = p \cdot q$ and e, d s.t. $e \cdot d = 1 \ (\varphi(n))$

(2) CRH $H: \mathcal{M} \to \mathbb{Z}_n$

output $pk = (n, e, H)$; $sk = (n, d, H)$

$S(sk, m \in \mathcal{M})$: $\sigma \leftarrow [H(m)^d \bmod n]$

$V(pk, m, \sigma)$: accept iff $[\sigma^e \bmod n] = H(m)$.

Problem: range of H depends on n. Inconvinient.

note: $e = 3 \Rightarrow$ superfast verify

why hash the msg ??

Answer: insecure if not!

Suppose $S(sk, m) = F^{-1}(sk, m) = [m^d \bmod n]$ for $m \in \mathbb{Z}_n$.

attack 1: on FDH sigs $F(pk, \cdot): X \to X$

choose $\sigma \in X$, set $m = F(pk, \sigma) = (\sigma^e \bmod n)$

output forgery (m, σ).

clearly $F(pk, \sigma) = m \implies$ exist. forgery!

hash prevents this: adv. needs to H-invert $F(pk, \sigma)$.

attack 2: blinding attack on RSA.

$\mathcal{A}$ wants sig on $m \in \mathbb{Z}_n$.

(1) choose $r \xleftarrow{R} \mathbb{Z}_n$, compute $\hat{m} = r^e \cdot m \pmod n$

(2) request sig. on $\hat{m} \in \mathbb{Z}_n$,

get back $\hat{\sigma}$ s.t. $(\hat{\sigma})^e = \hat{m} \pmod n$

(3) set $\sigma := \hat{\sigma}/r \pmod n$.

claim: σ is a valid sig. on m

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

$\Rightarrow$ can ask Alice to sign random $\hat{m}$, and get back sig on m] blind sigs.

- has apps to anonymous cash & voting.

<u>standard sigs</u>: PKCS1 v1.5

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

$$PKCS1(m) = \Big[\underbrace{\underset{16\text{ bits}}{\boxed{01}} \boxed{FF} \boxed{FF} \cdots \boxed{FF} \boxed{00} \boxed{SHA256(m)}}_{\text{RSA modulus size (2048 bits)}} \Big]$$

So: $\sigma = [PKCS1(m)^d \bmod n]$

<u>Problem</u>: this hash function is <u>not full domain</u>.

$\Rightarrow$ no security analysis.