Digital Signatures

Proj draft (optional) by Fri
HW7 coming, but lowest HW dropped

	requires shared secrets — symmetric key	no previously shared secrets required — public key
hide information — privacy	ENCRYPTION	ENCRYPTION
integrity/authenticity — guarantee the source of data	MAC	$\boxed{\text{Signatures}}$

Digital Signatures:

- KeyGen $\longrightarrow$ (vk, sk)
  - sk: (private) signing key
  - vk: (public) verification key
- Sign(sk, m) $\longrightarrow$ σ (sigma) — signature
- Ver(vk, m, σ) $\longrightarrow$ 0/1

Security: similar to MAC

"Hard to generate a forgery, even after seeing signatures of chosen messages"

forgery: (m, σ) that verify, but Adv never asked for Sig on m

Formal Definition

```
(vk, sk) ← KeyGen

GETVK():
    return vk

SIGN(m):
    return Sign(sk, m)

VER(m, σ):
    return Ver(vk, m, σ)
```

$\approx$

```
(vk, sk) ← KeyGen
S = ∅

GETVK():
    return vk

SIGN(m):
    σ ← Sign(sk, m)
    add (m, σ) to S
    return σ

VER(m, σ):
    return (m, σ) ∈? S
```

Weird:

Calling program doesn't need library's help to Verify

"Textbook" RSA signatures:

$N = pq$, $\qquad ed \equiv_{\phi(N)} 1$

public/verification	key	(N, e)
private/signing	key	(N, d)

$$\mathrm{Sign}\big((N,d), m\big) = \underline{m^d} \bmod N$$

$$\mathrm{Ver}\big((N,e), m, \sigma\big):$$

if $\sigma^e \equiv_N m$ then 1 else 0

Insecure: Idea: $\underbrace{m \mapsto m^d}_{\text{RSA func}}$ is a permutation

$\Rightarrow$ every σ is a valid signature of something

In particular, σ is valid signature of σ^e

Attack: get (N, e)
choose σ arbitrarily
set $m = \sigma^e \pmod N$
now (m, σ) is a forgery

Fix: Sign only $H(m)$ where H is hash function

$$\text{Sign}((N,d), m) = H(m)^d$$

(full-domain hash RSA)

Rabin Signatures

Obs #∅: If p odd prime then half of $\mathbb{Z}_p^*$ are squares

Ex: $p = 13$ $\quad \mathbb{Z}_p^* = \{1, \dots 13\}$

$1^2 = 1$	$5^2 = 12$	$9^2 = 3$
$2^2 = 4$	$6^2 = 10$	$10^2 = 9$
$3^2 = 9$	$7^2 = 10$	$11^2 = 4$
$4^2 = 3$	$8^2 = 12$	$12^2 = 1$

$\Rightarrow$ 6 squares out of 12 in $\mathbb{Z}_{13}^*$

<u>Obs #1:</u> If p odd prime and x is square mod p then

$$x^{(p-1)/2} \equiv_p 1$$

<u>Proof:</u> $x = y^2$ so

$$x^{(p-1)/2} = (y^2)^{(p-1)/2} = y^{\overset{\phi(p)}{p-1}} \equiv_p 1$$

<u>Obs #2:</u> If p is prime & $p \equiv_4 3$ and x is square mod p

then $x^{\frac{p+1}{4}}$ is a square root of x

why? $$\left(x^{\frac{p+1}{4}}\right)^2 = x^{\frac{p+1}{2}} = x^{1+\frac{p-1}{2}}$$

$$= x\left(x^{\frac{p-1}{2}}\right) = x$$

when I square this I get x

<u>Obs #3:</u> If N is RSA modulus, then $1/4$ of $\mathbb{Z}_N^*$ are squares

$N = 3 \cdot 5$ $\mathbb{Z}_N^* = \{1\ 2\ 4\ 7\ 8\ 11\ 13\ 14\}$

squares

square $\downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow\ \downarrow$

$1\ 4\ 1\ 4\ 4\ 1\ 4\ 1$ } 2 of 8

Obs #4: If you know p & q, and x is square mod pq then you can compute a sqrt of x

How? CRT

Obs #5: If you have a way to compute sqrts mod pq, then you can factor N

Idea: Suppose $Algo(x, N)$ returns $y : y^2 \equiv_N x$

How to factor N?

pick random r

set $x \equiv_N r^2$

$\Rightarrow$ x has 4 square roots

I know 2 of them: $\pm r$

call others $\pm s$

call $Algo(x, N)$

$\to$ w/ prob $\frac{1}{2}$ Algo outputs one of $\pm s$

Now I know r, s

$$r \not\equiv_N -s$$

$$r^2 \equiv_N s^2$$

$$\Rightarrow \quad r^2 - s^2 \equiv_N 0$$

$$(r+s)(r-s) \equiv_N 0$$

$\Rightarrow \gcd(r \pm s, N) = $ factors of N

RABIN SIGS:

verification key: $N = p \cdot q$ where $p \equiv_4 q \equiv_4 3$

signing key: p, q

Sign$((p,q), m)$:

choose random r

retry until $m \| r$ is a square mod N

compute u s.t. $\underbrace{u^2 \equiv_N m\|r}_{\text{compute Sqrt mod } N}$

return (r, u)

Verify$(N, m, (r,u))$:

check $u^2 \equiv_N m \| r$