Lecture 7: October 15, 2001

- Review: One-time sigs
- Rabin's OTS
- Public key cryptography
  - RSA
  - Etc.

- Project proposals due
- Last day to drop
- HW0 due Wednesday (October 17)
- HW1 due next Monday (October 22)
- Midterm: Wednesday (October 24)

"Optimal" One-time Signature of an n-bit message

$m - input$

$h - message\ di\ gest,\ h = hash(m)$

$n = |h|\ or\ \ log\,(max(h))$

$w = log(n) - 1$

$z = min(\#1bits\,(h), \#0bits\,(h))$

$SK = X_0, X_1, \{x_0, \ldots, x_{n/2-1}\}, \{x'_{0,0}, x'_{0,1}, \ldots, x'_{w-1,0}, x'_{w-1,1}\}$

$PK = Y_0, Y_1, \{y_0, \ldots, y_{n/2-1}\}, \{y'_{0,0}, y'_{0,1}, \ldots, y'_{w-1,0}, y'_{w-1,1}\}$

$f(x) = y$ (for all x, y)

$Signature : [X_0 \mid X_1, \{x_{i_0}, \ldots, x_{i_{z-1}}\}, \{x'_{0,0}, x'_{0,1}, \ldots, x'_{w-1,0}, x'_{w-1,1}\}]$

One-time Signature of n-bit message

Example:

$$h = 0100110100011000$$

$$n = 16, w = 3, z = 6 = 110$$

$$SIG(m) = m, X_1, (x_1, x_4, x_5, x_7, x_{11}, x_{12}), (x_{0,1}, x_{1,1}, x_{2,0})$$

Is it possible to forge a valid message?

One-time Signature Chains

How to sign n 2-bit messages

$$Y_{0,0} = \overbrace{f(\ldots f(x_{0,0})\ldots)}^{n \ times}$$

$$Y_{0,1} = f(\ldots f(x_{0,1})\ldots)$$

$$Y_{1,0} = f(\ldots f(x_{1,0})\ldots)$$

$$Y_{1,1} = f(\ldots f(x_{1,1})\ldots)$$

Publish all Y values

Suppose : $M1 = 01, M2 = 11, \ldots, Mn = 00$

$$Sig(M1) = \overbrace{f(\ldots f(x_{0,1})\ldots)}^{n-1 \ times}, \overbrace{f(\ldots f(x_{1,0})\ldots)}^{n-1 \ times}$$

$$Sig(M2) = \overbrace{f(\ldots f(x_{0,1})\ldots)}^{n-2 \ times}, \overbrace{f(\ldots f(x_{1,1})\ldots)}^{n-2 \ times}$$

...

$$Sig(Mn) = x_{0,0}, x_{1,0}$$

Example app: e-coins...
Used in S/Key (OTP, Opie)

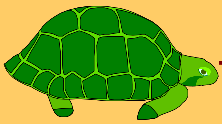
Rabin's One-Time Signatures

$K_i - random \;\; secret\; keys \;\; (0 < i < 2n)$

$Signature: \;\; S_i = E(K_i, H(MSG))$

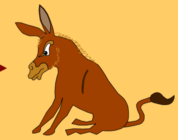
$Public-keys: \;\; P_i = E(K_i, i)$

$|i| = l \quad E - block - size$



$MSG, \;\; \Sigma = \{S_i \mid 0 < i < 2n\},$

$\mathrm{P} = \{P_i \mid 0 < i < 2n\},$



$Pick\; n\; random\;\; numbers:$

$R_i, \;\; 0 < i < 2n$

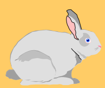
$$\Omega = \{R_i \mid 0 < i < 2n\}$$

$$\{K_{R_i} \mid R_i \in \Omega\}$$

$\forall R_i \in \Omega$

$D(K_{R_i}, S_{R_i}) \stackrel{?}{=} H(MSG)$

$D(K_{R_i}, P_{R_i}) \stackrel{?}{=} i$

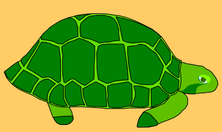


- **n – security parameter**
- **Inefficient**
- **On-line**

Rabin's One-Time Signatures (contd)

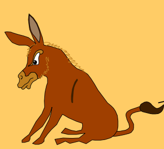
DISPUTES

Has P



$$\{K_i \mid 0 < i < 2n\}$$

$MSG, \;\; \Sigma$



1) $D(K_i, P_i) \stackrel{?}{=} i$

if not, Bob wins

2) $E(K_i, H(MSG)) \stackrel{?}{=} S_i$

for at most n S_i

Alice wins, else Bob wins

Note: Alice, Bob communicate with Court via authentic channels

Public Key Cryptography

- Merkle
- Hellman
- Diffie
- Rivest
- Shamir
- Adleman
- Rabin
- McEliece
- ElGamal

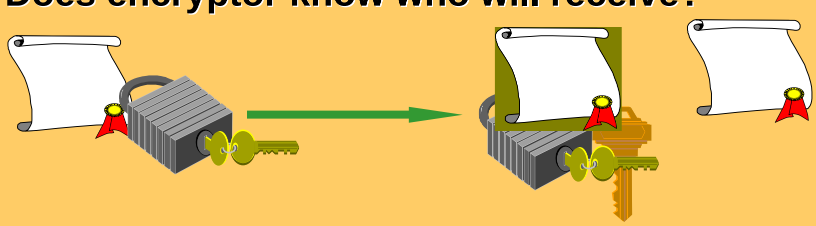
People

Cryptosystems

- RSA
- Diffie-Hellman KE
- Rabin
- El Gamal
- Elliptic Curve Duals
- Gillou-Quisquater
- Fiat/Shamir
- Knapsacks
- etc., etc.

What's the main idea?

- Everyone can encrypt for you
- No one can decrypt but you
- No pre-established pairwise secret info
- How is this different from conventional (shared-key) encryption?
- Do you know who encrypts for you?
- Does encryptor know who will receive?



RSA (1976-8)

$$
\begin{aligned}
&Let \quad n = pq \quad where \quad p \neq q - (large) \quad primes \\
&e, d \in_R Z_n \quad and \quad ed \equiv 1 \bmod \Phi(n) \\
&note \quad that : \Phi(n) = (p-1)(q-1) \\
&Secrets : p, q, d \\
&Publics : n, e \\
&Encryption : message = m \\
&E(x) = y = m^e \bmod n \\
&Decryption : ciphertext = y \\
&D(y) = x' = y^d \bmod n
\end{aligned}
$$

Why does it all work?

$$
\begin{aligned}
&x \in Z_n^* \\
&x^{ed} = x^{1 \bmod Ö(n)} \bmod n = \\
&x^{c*Ö(n)+1} \bmod n = x \\
&\text{But, recall that :} \\
&g^{Ö(n)} = 1 \bmod n \quad \text{(Lagrange)}
\end{aligned}
$$

How does it all work?

Example: p=17 q=13 n=221 (p-1)(n-1)=192=3^4*2

pick e=5, d=77 Can we pick 16? 9? 27? 185?

x=5, E(x)=3125 mod 221 = 31

D(y)=31^{77}=6.8367614277544200019639559955 8e+114 mod 221 =5

Example: p=5 q=7 n=35 (p-1)(n-1)=24=3*2^3

pick e=11, d=11

x=2, E(x)=2048 mod 35 =18=y

y=18, D(y)=6.426841007923e+13 mod 35 = 2

Why is it secure?

Conjecture: breaking RSA is *polynomially equivalent* to factoring n. Recall that n is very, very large!

Why: n has unique factors p,q

Given p,q computing (p-1)(q-1) is easy:

$$ed \equiv 1 \, mod \, \Phi(n)$$

Use extended Euclidean!

Exponentiation Costs

- ❑Integer multiplication -- $O(b^2)$ b -- bit length of base m
- ❑Modular reduction -- $O(b^2)$
- ❑Thus, modular multiplication -- $O(b^2)$
- ❑Modular exponentiation -- m^e mod n
- ❑Naïve method: e-1 modular products -- $O(b^2*e)$

BUT e is large, as large as?

- ❑Let L= |e| (e.g., L=1024 for 1024-bit RSA exponent)
- ❑We can assume b and L are close in length
- ❑Square-and-multiply method works in $O(b^3)$ time… $O(b^2*2L)$

Square-and-Multiply

```
compute m^e mod n
--------------
l = sizeof(n);
temp = 1;
 for (i = l-1; i >= 0; i--)
{   temp* = temp;
    temp % = n;
    if (e[i])
    {   temp* = m;
        temp% = n;
    }
}
```

From left to right in e

- •Example 1: e=100
- •Example 2: e=10000000
- •Example 3: e=11111111

N=35, e=11, m=2

Speeding it up

Let :

$d_p = d \bmod (p-1)$

$d_q = d \bmod (q-1)$

compute :

$M_p = C^{d_p} \bmod p$

$M_q = C^{d_q} \bmod q$

and solve :

$M = M_p \bmod p$

$M = M_q \bmod q$

$$M = [M_p q (q^{-1} \bmod p)$$
$$+ M_q p (p^{-1} \bmod q)] \bmod (pq)$$