Lecture 9: October 22, 2001

- El Gamal PKCS
- Signatures:
  - RSA
  - El Gamal
  - DSS

- HW1 due today
- Midterm: Wednesday (October 24)

El Gamal and Discrete Logs

The discrete logarithm problem in Zp
(where p - large prime, at least 512 bits)

Eve knows:
p, b, y
$p - prime$
$b \in Z_p^* - generator, base$
$y \in Z_p^* - query\ element$
$FIND\ x \in [0, p-2]\ \ni\ b^x \equiv y \bmod p$

Factoids:
- no known poly algorithms
- basis for many crypto methods
- conjectured to be a good OWF; why?

El Gamal PK cryptosystem (83)

p – large prime

b – base, primitive element, generator

x – private exponent

y – public residue; $y \equiv b^x \bmod p$

$P = Z_p^*$

$C = Z_p^* \times Z_p^*$

publics : p, b, y

secrets : x

Encryption:

1. generate random $r \in Z_{p-1}$
2. compute: $k = b^r \bmod p$
3. compute: $c = my^r \bmod p = mb^{xr} \bmod p$
4. ciphertext $= \{k, c\}$

Decryption:

1. compute $k^x \bmod p$
2. compute $(k^x)^{-1} \bmod p$
3. $m' = (k^x)^{-1} c = b^{-rx} mb^{xr} \bmod p = m$

El Gamal (example)

$p = 13, b = 2, x = 9$

$y = 2^9 \bmod 13 = 5$

Encryption:

$m = 11$

$r = 10$

$k = 2^{10} \bmod 13 = 10$

$c = 11 * 5^{10} \bmod 13 = 2$

$ciphertext = \{10, 2\}$

Decryption:

$10^9 \bmod 13 = 12$

$12^{-1} \bmod 13 = 12$

$2 * 12 = 24 \equiv 11 \bmod 13$

Merkle-Hellman Knapsack PKCS

Both are NP-c

Subset sum decision problem

$S = \{s_1,...,s_n\} - sizes$
$T - target\ sum$

Does there exist a binary vector
$X = \{x_1,...,x_n\}$
such that :
$\sum S \times X = T$

Subset sum search problem

$S = \{s_1,...,s_n\} - sizes$
$T - target\ sum$

Compute a binary vector
$X = \{x_1,...,x_n\}$
such that :
$\sum S \times X = T$

$S = \{s_1,...,s_n\}$
superincreasing
iff $\forall i$
$s_i > s_1 + ... + s_{i-1}$

→ Trivial to solve...

Merkle-Hellman (contd)

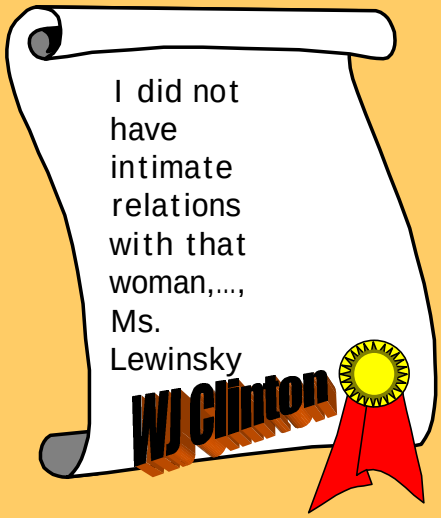
$S = \{s_1,...,s_n\} - sizes\ (superincreasing)$
$p - prime,\ p > \sum S$
$a < p$
$C = \{c_1,...,c_n\}$
$c_i = as_i \bmod p$

publics : C
secrets : p, a, S

Encryption :
cleartext $X = \{x_1,...,x_n\}$
$e_k(X) = \sum XC$

Decryption :
ciphertext Y
1. $T = a^{-1}Y \bmod p$
2. *solve SSP for* $S, T -> X$

Digital Signatures



- Integrity
- Authentication
- Non-repudiation
- Timestamping
- Causality
- Authorization

Digital Signatures

A signature scheme:

(P,A,K,Sign,Verify)

***P* - plaintext (msgs)**

***A* - signatures**

***K* - keys**

Sign* - signing function: (*P*K)->A

Verify* - verification function: *(P*A*K)->{0,1}

RSA Signature Scheme

$Let\ n = pq\ where\ p \neq q - (large)\ primes$

$e, d \in_R Z_n \ \ and\ e = d^{-1}\ and\ ed \equiv 1\ mod\ \Phi(n)$

$\Phi(n) = (p-1)(q-1)$

$Secrets: p, q, d$

$Publics: n, e$

$Signing: message = m$
$Sign(x): y = m^d\ mod\ n$

$Verification: signature = y$
$Verify(y, m): (m = y^e)???$

RSA Signature Scheme (contd)

The good:
- •Verification can be made cheap
- •Same as decryption function
- •Security based on RSA encryption
- •Signing is harder but #verify-s > 1…
- •Deterministic

The bad:
- •Recall that RSA is malleable: signatures can be "massaged"
- •Phony "random" signatures
  - •compute Y=RSA(e,X)=X^e mod n
  - •X is a signature of Y because Y^d=X mod n

The ugly:
- •Signing requires integrity!
- •How to sign multiple blocks?
- •Deterministic

El Gamal Signature Scheme

p – large prime
b – base, generator
x – private exponent
y – public residue ; $y \equiv b^x \bmod p$
$P = Z_p^*$
$A = Z_p^* \times Z_p^*$
publics : p, b, y
secrets : x

Signing :
1. generate random $r \in Z_{p-1}$
2. compute : $k = b^r \bmod p$
3. compute : $c = (m - xk)r^{-1} \bmod p - 1$
4. signatur $e = \{k, c\}$

Verifying :
$y^k k^c \bmod p = b^m \bmod p$???

notice that :
$y^k k^c = b^{xb^r} (b^r)^{(m/r - xk/r)} = b^{xb^r + m - xb^r} = b^m$