Public-Key Encryption

Ramya Pradhan

3/20/2008

Agenda

- Idea
- Security
- RSA Cryptosystem
- Rabin Encryption
- Diffie-Hellman Key Exchange
- ElGamal Encryption

Idea

- Symmetric Cryptosystems
  - Key distribution
  - Key management
- Public-key Cryptosystems
  - Decryption key or Private key
  - Encryption key or Public key
- Computing private keys from corresponding public key – infeasible!
- Symmetric Cryptosystems more efficient than Public-key Cryptosystems

Security

- Security of the secret key
  - computing secret key from publicly available information is intractable
- Semantic security
  - a passive attacker with limited resources cannot obtain information about the plaintext that corresponds to a given cipher text
- Chosen cipher text security
  - the attacker can decrypt any cipher text of his choice except for the cipher text he is really interested
- Security proofs
  - Intractability of computational problems

RSA Cryptosystem

- Key Generation
- Encryption
- Decryption
- Security of the Secret Key
- RSA and factoring
- Choice of p and q
- Choice of e
- Choice of d
- Efficiency
- Multiplicativity
- Secure RSA
- Generalization

RSA Key Generation

Steps:

1. Large prime numbers p and q
2. RSA modulus n = pq
3. Encryption exponent e,
   $1<e<\phi(n)=(p-1)(q-1)$ & $\gcd(e,(p-1)(q-1)) = 1$
4. Decryption exponent d,
   $1 < d < (p-1)(q-1)$ & $de = 1 \bmod (p-1)(q-1)$

Keys:

- Public key : pair(n, e)
- Private key: d

RSA Encryption

- Plain text m, $0 \leq m < n$
- Cipher text c, $c = m^e \bmod n$

Example:

Let p = 11 and q = 23. Then, n = 253 and e = 3. The plain text space = {0, 1,…,252}.

If m = 26, then $c = 26^3 \bmod 253 = 119$

RSA Decryption

- Theorem

  Let (n,e) be a public RSA key and d the corresponding private RSA key, then

  $$(m^e)^d \bmod n = m$$

  for any integer m with $0 \leq m < n$

Example:

Our n = 253, e = 3 and d = 147 and c = 119.

$119^{147} \bmod 253 = 26$

Security of the RSA secret key

- Factoring problem for integers
- Suppose the attacker knows p and q, then d may be computed by solving the congruence

  $de \equiv 1 \bmod (p-1)(q-1)$
- The converse is also true i.e. p and q can be computed from n, e, d

  $s = \max\{t \in N: 2^t \text{ divides } ed-1\}$

  and $k = (ed - 1)/2^s$

Factorization of n

- Lemma

  For all integers a that are prime to n, the order of the residue class $a^k + nZ$ in the group $(Z/nZ)^*$ is in $\{ 2^i : 0 \leq i \leq s\}$

- Theorem

  Let a be an integer that is prime to n. If the orders of the residue class a^k+pZ in $(Z/pZ)^*$ and of a^k+qZ in $(Z/qZ)^*$ are different, then $1 < \gcd(a^{2^t k} - 1, n) < n$ for some $t \in \{0,1,\ldots,s-1\}$

- Theorem

  The number of integers a prime to n in the set $\{1,2,\ldots,n-1\}$ for which a^k has a different order mod p and mod q is at least

  $(p-1)(q-1)/2$

Factoring n:

Steps:

1. Choose at random an integer a in the set {1,…,n-1}
2. Compute g = gcd(a, n)
3. If g = 1, then compute gcd($a^{2^{tk}} - 1 \bmod n$, n) for t = s-1, s-2,… until g > 1 or t=0
4. If g > 1, then g = p or g = q

Example :

Let n = 253, e = 3 and d = 147. Hence, ed-1 = 440. If we use a = 2, then we obtain gcd($2^{220} - 1$, 253) = gcd($2^{110} - 1$,253) = gcd($2^{55} - 1$,253) = 23

Our initial q value!

RSA and factoring

- Is breaking RSA as difficult as factoring integers?
- Is RSA secure?
- Or, is factoring difficult?

Choice of p and q

- Random primes
- Almost of equal length and at least 512 bits long

Choice of e

- As small as possible , $e \geq 3$
- Low-exponent attack!

Choice of d

- Small value for efficient computation of e
- Store on smart card

RSA Efficiency

- Encryption and decryption require one exponentiation modulo n, each
- Encryption exponent e – as small as possible
- Decryption exponent d – as large as n
- If RSA modulus n is k bits, then decryption requires k squarings and k/2 multiplications mod n
- Chinese remainder theorem – Faster decryption

Decryption using Chinese remainder theorem:

1. Private key d
2. Compute
   $m_p = c^{d \bmod p-1} \bmod p$, $m_q = c^{d \bmod q-1} \bmod q$
3. Compute $m \in \{0,1,\ldots,n-1\}$ such that
   $m = m_p \bmod p$, $m = m_q \bmod q$
4. Use Extended Euclidean algorithm to compute
   $y_p p + y_q q = 1$
5. Compute $m = (m_p y_q q + m_q y_p p) \bmod n$

Multiplicativity

- Let m_1 and m_2 be two messages, then $c_1 = m_1^{\,e} \bmod n$ and $c_2 = m_2^{\,e} \bmod n$
- The product $c = c_1 c_2 \bmod n = (m_1 m_2)^{e} \bmod n$
- If c_1 and c_2 are known, then encryption of $m = m_1 m_2$ can be computed
- Existential forgery

Secure RSA

- Not semantically secure
- Insecure against chosen cipher test attack
- Solution – Randomize RSA!
- RSA variant based on Optimal Asymmetric Encryption Protocol (OAEP)
- Idea:
  - Construct random functions using cryptographic hash functions
  - Mask a random number r
  - Randomize the plain text and r using the functions

Generalization

- Public key – finite group G
- Encryption exponent e – prime to the order o of G
- RSA $G = (Z/nZ)^*$, n – RSA modulus
- Secret key d, $de \equiv 1 \bmod o$
- Encryption of $m \in G$, $c = m^e$
- Decryption $c^d = m^{ed} = m$
- Factoring integers

Rabin Encryption

- Key generation
- Encryption
- Decryption
- Efficiency
- Security against cipher-text only attacks
- A chosen cipher text attack
- Secure Rabin Encryption

Rabin Key Generation

- Large prime numbers p and q, $p \equiv q \equiv 3 \bmod 4$
- $n = pq$
- Public key – n
- Private key pair – (p,q)

Rabin Encryption

- $m = \{0, 1, \ldots, n-1\}$
- $c = m^2 \bmod n$

Rabin Decryption

- Extract square roots of c
- $m_p = c^{(p+1)/4} \bmod p$, $m_q = c^{(q+1)/4} \bmod q$
- $\pm m_p + pZ$ two square roots of $c + pZ$
- $\pm m_q + qZ$ two square roots of $c + qZ$
- Compute 4 square roots of $c + nZ$ using Chinese Remainder Theorem
- Use Extended Euclidean to obtain $y_p p + y_q q = 1$
- Compute $r = (y_p p\, m_q + y_q\, q m_p) \bmod n$
  and $s = (y_p p\, m_q - y_q\, q m_p)$
- $m = \pm r$ or $\pm s$

Rabin Encryption Efficiency

- Encryption requires only one squaring
- Decryption expensive

Cipher text only attack

- Factoring Rabin modulus = Breaking Rabin system
- Converse is also true

Example:

Let n = 253. Let square roots modulo 253 be computed using algorithm R. If plain text x = 17 is chosen, then gcd(17, 253) = 1

$c = 17^2 \bmod 253 = 36$

Square roots of 36 mod 253 = 6, 17, 236, 247

gcd(6-17, 253) = 11 and gcd(247-17, 253) = 23

If R yields either of these, then n has been successfully factored!

A chosen cipher text attack

- Decrypt text of one's choice
- Factoring of Rabin modulus

Solution:

- Reduce plain text to special form
- Breaking Rabin system and factoring Rabin modulus equivalence is lost

Secure Rabin Encryption

- Constructed similar to Secure RSA Encryption

Diffie-Hellman Key Exchange

- Exchanging keys over insecure channels
- Basis for ElGamal Cryptosystem
- Information obtained during key exchange cannot be used to construct secret key
- Discrete Logarithm Problem

Discrete Logarithms

- Prime number p
- Group (Z/pZ)* is cyclic of order p-1
- g, Primitive root mod p
- A $\in$ {1, 2, …, p-1}, then a $\in$ {0, 1, 2, …, p-2} with $A \equiv g^a \bmod p$
- a $\longrightarrow$ discrete logarithm of A to the base g
- $a = \log_g A$

Example:

- p = 13, g = 2 (primitive modulo 13)
- Discrete logarithms of {1, 2, …, 12} to base 2 are

A	1	2	3	4	5	6	7	8	9	10	11	12
$\log_2 A$	0	1	4	2	9	5	11	3	8	10	7	6

- Consider A = 5. Its corresponding $\log_2 A$ is 9, that is $5 \equiv 2^9 \mod 13$ or $9 = \log_2 5$, i.e.
  $2^9 = 512 \equiv 5 \mod 13$

Key Exchange between Alice & Bob

- Large prime number p
- Integer g, $2 \leq g \leq p-2$ (i.e. primitive root)
- Alice chooses $a \in \{0, 1, \ldots, p-2\}$ randomly, computes $A \equiv g^a \bmod p$, keeps a secret
- Sends A to Bob
- Bob chooses $b \in \{0, 1, \ldots, p-2\}$ randomly, computes $B \equiv g^b \bmod p$, keeps b secret
- Sends B to Alice
- Alice computes $B^a \bmod p = g^{ab} \bmod p$
- Bob computes $A^b \bmod p = g^{ab} \bmod p$
- Common key $K = g^{ab} \bmod p$

Example:

- Let p = 17 and g = 3.
- Alice's a = 7 . Thus, A = 3^7 mod 17 = 11
- Sends A = 11 to Bob
- Bob's b = 4. Thus, B = 3^4 mod 17 = 13
- Sends B = 13 to Alice
- Alice computes 13^7 mod 17 = 4
- Bob computes 11^4 mod 17 = 4
- Therefore, common key K = 4

Selection of g

- Integer g whose order mod p is sufficiently high
- Primitive root mod p
- Residue class g + pZ has high order in group (Z/pZ)*

Higher order mod p ensures:

- Factoring of (p-1) is intractable
- Hence, finding primitive root of p is also intractable

Security:

- Diffie-Hellman Problem
  - Oscar knows p, g, A and B
  - Oscar does not know a and b
  - Compute $K = g^{ab} \bmod p$ – Diffie-Hellman problem
  - If b is computed, then $K = A^b \bmod p$
- Decision Diffie-Hellman Problem
  - Given $g^a \bmod p$, $g^b \bmod p$ & $g^c \bmod p$
  - Is $g^c \bmod p = g^{ab} \bmod p$?
- Man in the middle attack
  - Digital Signatures

ElGamal Cryptosystem

- Based on solving Diffie-Hellman problem in (Z/pZ)*

Key Generation

- Choose a prime number p
- Choose a primitive root g mod p
- Choose random exponent $a \in \{0, \ldots, p-2\}$
- Compute $A = g^a \bmod p$

Keys:

- Public key (p, g, A)
- Fixed Private key (a)

ElGamal Encryption

- Plain text space – {0, 1, …, p-1}
- Bob encrypts plain text m using Alice's (p, g, A)
- Bob chooses b $\in$ {1, …, p-2} and computes
  - $B = g^b \bmod p$
  - $c = A^b \, m \bmod p$

Cipher text: (B, c)

ElGamal Decryption

- Alice has (B, c)
- Divide c by $B^a \bmod p$
- Compute $x = p-1-a$, to avoid inversion mod p
- Compute $m = B^x c \bmod p$

This works because:

$$B^x c \equiv g^{b(p-1-a)} A^b m \equiv (g^{p-1})^b (g^a)^{-b} A^b m \equiv m \bmod p$$

Example:

Alice's p = 23, g = 7, a = 6 and $A = g^a \bmod p = 4$.

Her public key : (23, 7, 4) and secret key: (6)

Bob encrypts m = 7. Bob's b = 3.

He computes $B = g^b \bmod p = 21$

He computes $c = A^b \; m \bmod p = (4^3 \cdot 7) \bmod 23 = 11$

Cipher text: (21, 11)

Alice decrypts by computing

$B^{p-1-6} \; c \bmod p = (21^{16} \cdot 11) \bmod 23 = 7 = m!$

Efficiency

- Two modular expansions
  - Precomputations
- One modular multiplication

ElGamal and Diffie-Hellman

If Diffie-Hellman is broken, then ElGamal can be broken and vice-versa.

Choice of parameters

- Randomly choose prime number p with equal distribution from all primes of certain length

- If messages m_1 and m_2 are encrypted as c_1 and c_2 using same exponent b, then $c_1^{-1} c_2 = m_1^{-1} m_2 \bmod p$

  If m is known,

  $m_2 = c_1^{-1} c_2 m_1 \bmod p$

  Therefore, choose new b for every new encryption

Generalization

- ElGamal cryptosystem can be implemented in any cyclic group
  - Efficient computations
  - Difficult to solve Diffie-Hellman problem
  - Infeasible to compute discrete logarithms

- If efficient DL algorithm for $(Z/pZ)^*$ is found, then switch to $(Z/p` Z)^*$ for which the problem still exists ☺

Questions

- RSA Cryptosystem
- Rabin Encryption
- Diffie-Hellman Key Exchange
- ElGamal Cryptosystem

Thank you!