CIS 5371 Cryptography

8. Encryption -- Asymmetric Techniques

Textbook encryption algorithms

In this chapter, security (confidentiality) is considered in the following sense:

- *All-or-nothing secrecy*. Given the ciphertext $y=E_K(x)$, of plaintext x, the task of the attacker is to retrieve the whole of x. Otherwise he fails.
  - That is, the adversary either gets x or nothing.
  - Nothing means that the attacker does not have any knowledge about x before of after the attack.
- *Passive attacker.* The attacker does not manipulate or modify the ciphertext using data she/he has in possession and does not ask a key owner to provide encryption or decryption services.

Textbook encryption algorithms

- *All-or-nothing secrecy*.

  In applications, plaintext data is likely to have partial information known to the attacker.

- *Passive attacker.*

  One should never expect an attacker to be so nice and remain passive.

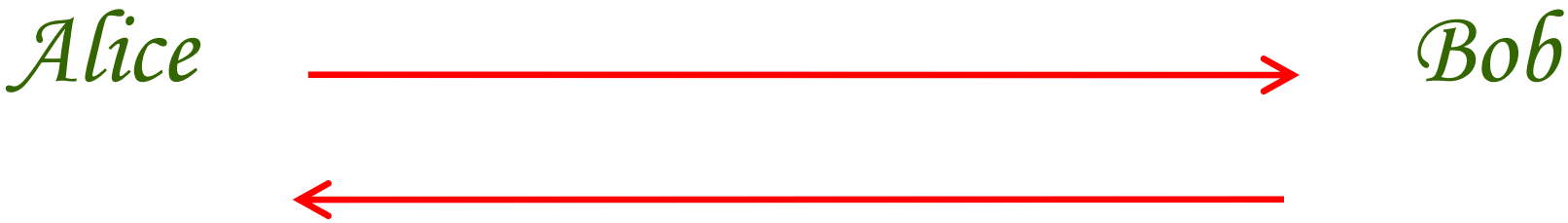
Textbook encryption algorithms security paradigms

We regard the security offered by the

- *All-or-nothing secrecy*, and the
- *Passive attacker*

as models for our *security paradigms*.

Public Key Cryptography



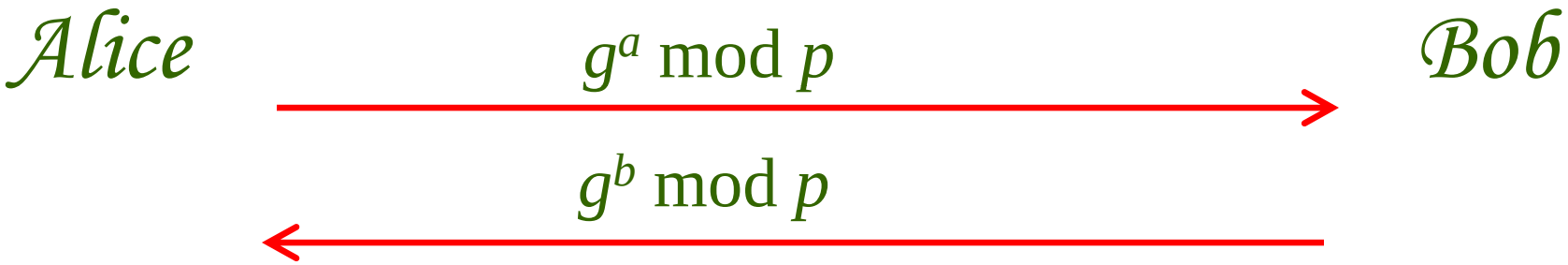
Alice and Bob want to exchange a private key in public.

Public Key Cryptography
The Diffie-Hellman protocol

Let p is a *large* prime and $g \in Z_p^*$ an element that generates a group of *large* prime order.

- The order q of g is a factor of p-1.
- If $q = p$-1, then we say that g is a generator of Z_p^*
- Usually we take q to be a prime.

Public Key Cryptography
The Diffie-Hellman protocol



The private key is: $g^{ab} \bmod p$

where p is a prime and g is a generator of Z_p^*

Example

- $p = 43$, $g = 3$,
- Alice and Bob share $(p,g) = (43,3)$.
- Alice picks at random her secret exponent $a = 8$
- Alice sends Bob: $3^8 \equiv 25 \pmod{43}$.
- Bob picks at random his secret exponent $b = 37$
- Bob sends Alice: $3^{37} \equiv 20 \pmod{43}$.
- The secret key agreed between the two is:

$$9 \equiv 20^8 \equiv 25^{37} \pmod{43}.$$

Man-in-the-middle attack

- Alice picks $a \in_U Z_p^*$ and sends Malice ("Bob"): $g^a \pmod p$
- Malice ("Alice") picks $m \in Z_p^*$ and sends Bob: $g^m \pmod p$
- Bob picks $b \in_U Z_p^*$ and sends Malice ("Bob) Bob: $g^b \pmod p$
- Malice ("Bob") sends Alice: $g^m \pmod p$
- Alice computes: $k_1 \leftarrow (g^m)^a \pmod p$
- Bob computes: $k_2 \leftarrow (g^m)^b \pmod p$

The Diffie-Hellman Problem

The Computational Diffie-Hellman Problem -- CDH

- INPUT
  - The description of a finite cyclic group G of order q (say Z_q^*)
  - A generator element g of G
  - $g^a, g^a \in G$, for some integers $0 < a,b < q$.
- OUTPUT
  - g^{ab}

The Diffie-Hellman Assumption

- A CDH solver is a PPT algorithm $\mathcal{A}$ that solves the CDH problem with advantage $\varepsilon > 0$.
- The DHA is that, for any $\varepsilon > 0$, and any arbitrary instance of the CDH problem, there is no CDH solver that will succeed for all sufficiently large inputs.

The Discrete Logarithm Problem

The Discrete Logarithm Problem -- DL

- INPUT
  - The description of a finite cyclic group G of order q (say Z_q^*)
  - A generator element g of G
  - $h \in G$.
- OUTPUT
  - The unique integer $a < q$ such that $h = g^a \bmod q$.

We call the integer a the discrete logarithm of h in base g and write: $a = \log_g h \bmod q$.

The Discrete Logarithm Assumption

- A DL solver is a PPT algorithm $\mathcal{A}$ that solves the DL problem with advantage $\varepsilon > 0$.
- The DLA is that, for any $\varepsilon > 0$, and any arbitrary instance of the DL problem, there is no DLA solver that will succeed for all sufficiently large inputs.

The RSA cryptosystem

Alice performs the following steps

- Choose p, q large primes with $|p| \approx |q|$.
- Compute $N = pq$.
- Compute $\varphi(N) = (p-1)(q-1)$.
- Choose a random integer $e < \varphi(N)$ such that $\gcd(e, \varphi(N)) = 1$ and compute the integer d such that $ed = 1 \pmod{\varphi(N)}$.
- Make public (N,e) as her public key, keep (N,d) as her private key, and discard p,q and $\varphi(N)$.

The RSA cryptosystem

Encryption

Let $m < N$ be the confidential message that Bob wants to send to Alice.

- Bob creates the ciphertext: $c \leftarrow m^e \pmod{N}$.
- Bob sends Alice: c

Decryption

To decrypt the ciphertext c Alice computes:

$$m \leftarrow c^d \pmod{N}.$$

Check

We have:

- $ed = 1 \pmod{\varphi(N)}$, so $ed = 1 + t\varphi(N)$.

Therefore,

- $D_d(E_e(m)) = (m^e)^d = m^{ed} = m^{t\varphi(N)+1}$

$$= (m^{\varphi(n)})^t \times m = 1 \times m = m \bmod n$$

Example

- Let $p = 101$, $q = 113$. Then $N = 11413$.
- $\varphi(N) = 100 \times 112 = 11200 = 2^6 5^2 7$
- For encryption use $e = 3533$.
- Alice publishes: $N = 11413$, $e = 3533$.
- Suppose Bob wants to encrypt: 9726.
- Bob computes $9726^{3533} \bmod 11413 = 5761$
- Bob sends Alice the ciphertext 5761.
- To decrypt it Alice computes the plaintext:

$$5761^{6597} \pmod{11413} = 9726$$

Implementation

1. Generate two large primes: p,q
2. $N \leftarrow pq$ and $\varphi(N)= (p-1)(q-1)$
3. Choose random e: with $1 < e < \varphi(N)$ & $\gcd(e,\varphi(N))=1$
4. $d \leftarrow e^{-1} \bmod f(n)$
5. The public key is (n,e) and the private key is (N,d)

Cost

In Z_n:

– Cost of a modular multiplication $(x \times y) \bmod n$ is

$$O(k^2), \text{ where } k = |\log_2 n|$$

– Cost of a modular exponentiation $x^z \pmod n$ is

$$O(k^2 \log_2 z)$$

Cryptanalysis of Public-key cryptosystems

Active attacks on cryptosystems

- Chosen-Plaintext Attack (CPA):
  - *The attacker chooses plaintexts and obtains the corresponding ciphertexts: the task of the attacker is successful if he can decrypt a (new) target ciphertext.*
- Chosen-Ciphertext Attack (CCA1):
  - *The attacker chooses a number of ciphertexts and obtains the corresponding plaintexts: the task of the attacker is successful if he can decrypt a (new) target ciphertext.*
- Adaptive Chosen-Ciphertext Attack (CCA2):
  - *This is a CCS1 attack in which the attacker can adaptively choose ciphertexts: the task of the attacker is successful if he can decrypt a (new) target ciphertext.*

The RSA Problem

The RSA Problem -- RSA

- INPUT
  - $N = pq$ with p,q prime nmbers.
  - e an integer such that gcd(e,(p-1)(q-1)) =1
  - $c \in Z_N$.
- OUTPUT
  - The unique integer $m \in Z_N$ such that $m^e \equiv c \pmod{N}$

The RSA Assumption

- An RSA solver is a PPT algorithm $\mathcal{A}$ that solves the RSA problem with advantage $\varepsilon > 0$.
- The RSA Assumption is that, for any $\varepsilon > 0$, and any arbitrary instance of the RSA problem, there is no RSA solver that will succeed for all sufficiently large inputs.

The Integer Factorization Problem

The IF Problem -- IF

- INPUT
  - N an odd composite integer with at least two distinct prime factors.
- OUTPUT
  - A prime p such that $p \mid N$.

The IF Assumption

- An integer factorizer is a PPT algorithm $\mathcal{A}$ that solves the IF problem with advantage $\varepsilon > 0$.
- The IF Assumption is that, for any $\varepsilon > 0$, and any arbitrary instance of the IF problem, there is no integer factorizer that will succeed for all sufficiently large inputs.

Security of RSA

1. Relation to factoring.

Recovering the plaintext *m* from an RSA ciphertext *c* is easy if factoring is possible.

2. The RSA problem

Recovering the plaintext *m* from an RSA ciphertext *c* is easy if the RSA problem is easy.

3. Relation between factoring and the RSA problem

- If Factoring is easy then the RSA problem is easy.
- The converse is likely not to be true.

The Rabin cryptosystem

Alice performs the following steps

- Choose p, q large primes with $|p| = |q|$.
- Compute $N = pq$.
- Pick a random integer $b \in_U Z_n^*$
- Make public (N,b) as her public key, keep (p,q) as her private key.

The Rabin cryptosystem

Encryption

Let $m \in Z_n^*$ be the confidential message that Bob wants to send to Alice.

- Bob creates the ciphertext: $c \leftarrow m(m+b) \pmod{N}$.
- Bob sends Alice: c

Decryption

To decrypt the ciphertext c Alice solves the quadratic equation:

$$m^2 + bm - c \equiv 0 \pmod{N},$$

for $m < N$.

The Rabin cryptosystem

Decryption

From elementary mathematics: $m \equiv \frac{-b+\sqrt{\Delta}}{2} \pmod{N}$,

where $\Delta = b^2 + 4c \pmod{N}$.

Since m was chosen in Z_n^*, Δ must be in QR_N.

Notice that if p,q are such that $p \equiv q \equiv 3 \pmod 4$, then it is easier to compute square roots modulo N.

Remarks

- Suppose $p \equiv q \equiv 3 \pmod 4$, $n = pq$.
- Let $y \equiv x^2 \pmod n$.
- Then:

$$(\pm y^{(p+1)/4})^2 \equiv y^{(p+1)/2} \equiv y^{(p-1)/2} \times y \equiv y \pmod p$$

Because

$$y^{(p-1)/2} \equiv 1 \pmod p .$$

Remarks (continued)

- It follows that:

$$\pm\, y^{(p+1)/4} \pmod{p}$$

  is a square root of y modulo p.
- A similar argument applies for the other prime q.
- So we get the quadratic residues modulo p and modulo q.
- We then get the quadratic residue modulo n by using the Chinese Remainder Theorem.

Example

Suppose n = 77.

Then $e(x) = x^2 \pmod{77}$

$d(y) = \sqrt{y} \pmod{77}$

Suppose Bob wants to decrypt y = 23.

$$\pm 23^{(7+1)/4} \equiv \pm 2^2 \equiv \pm 4 \bmod 7$$

$$\pm 23^{(11+1)/4} \equiv \pm 1^3 \equiv \pm 1 \bmod 11$$

Example, continued

Using the Chinese Remainder Theorem we compute the 4 square roots of 23 modulo 77 to be:

$$\pm 10 \pmod{77}, \quad \pm 32 \pmod{77}$$

The Rabin Problem

- INPUT
  - $N = pq$ with p,q prime numbers.
  - $y \equiv x^2 \pmod{N}$, $x \in Z_N^*$
- OUTPUT
  - $z \in Z_N^*$ such that $z \equiv x^2 \pmod{N}$.

Security of Rabin

1. Relation to factoring.

Recovering the plaintext *m* from a Rabin ciphertext *c* is easy if the IF problem is easy.

3. Relation between factoring and the Rabin problem

- Under CPA attacks the Rabin system is secure iff the IF problem is hard.
- Under CCA attacks the Rabin problem is completely insecure.

Security of Rabin

Under CPA attacks the Rabin system is secure iff the IF problem is hard.

Proof:

We show this for the case when $b = 0$.

Suppose that there is an algorithm that breaks Rabin with non-negligible probability $\varepsilon > 0$.

Let m be a random message, $c \equiv m^2 \pmod{N}$.

The decryption m' of m is one of the 4 square roots of c.

With probability ½, we have $m' \neq \pm m \pmod{N}$.

Then $\gcd(m' \pm m, N) = p$ or q.

This contradicts the IF assumption. The converse is trivial.

Security of Rabin

Under CCA attacks the Rabin system is completely insecure.

Proof:

We show this for the case when $b = 0$.

The adversary picks an m and computes $c \equiv m^2 \pmod{N}$.

Then he gets its decryption m'.

This is one of the 4 square roots of c, and with probability ½,

$$\gcd(m' \pm m, N) = p \text{ or } q.$$

Then the adversary can decrypt any ciphertext.

The ELGamal cryptosystem

Alice performs the following steps

- Choose a large random primes p.
- Compute a random multiplicative generator $g \in Z_p^*$
- Pick $x \in_U Z_{p-1}$ as private key
- Compute the public key $y \leftarrow g^x \pmod{p}$.
- Make public (p,g,y) as her public key, and keep (p,x) as her private key.

The ElGamal cryptosystem

Encryption

Let $m < p$ be the confidential message that Bob wants to send to Alice.

Bob picks $k \in_U Z_{p-1}$ and computes the ciphertext (c_1, c_2)

- $c_1 \leftarrow g^k \pmod{p}$.
- $c_2 \leftarrow y^k m \pmod{p}$.

Decryption

To decrypt the ciphertext (c_1, c_2) Alice computes

$$m \leftarrow c_2 / c_1^x \pmod{p}.$$

The ElGamal cryptosystem

Check:

$$c_1^{\,x} \equiv (g^k)^x \equiv y^k \equiv c_2/m \pmod{p}.$$

Example

Use $p = 43,\ g = 3,\ m = 14,\ x = 7,\ y = 37.$

Alice's private key: $x = 7$

Alice's public key: $(p,g,y) = (43,3,37)$.

Encryption with $k = 26$:

- $c_1 = g^k \pmod p = 3^{26} \pmod{43} = 15.$
- $c_2 = y^k m \pmod p = 37^{26} \times 14 \pmod{43} = 31$.

Decryption:

$$m = c_2 / c_1^x \pmod p = 31/15^7 \pmod{43} = 14 .$$

Security of ElGamal

1. Relation to the DL.

Recovering the plaintext *m* from an ElGamal ciphertext *c* is easy if the DL problem is easy.

2. ELGamal and the CDH problem

For messages that are uniformly distributed, the ElGamal encryption system is secure against CPAs iff the CDA problem is hard.

Security of ElGamal

For messages that are uniformly distributed, the ElGamal encryption system is secure against CPAs iff the CDH problem is hard.

Proof:

Suppose there exists an oracle that breaks ElGamal with non negligible probability $\varepsilon > 0$.

Then since $m = c_2/c_1^{\,x} \pmod p$ can be computed,

$$c_2/m = g^{(\log_g y \log_g c_1)} \pmod p$$ can also be computed.

For an arbitrary CDH instance (p,g,g_1,g_2), take (p,g,g_1) as the ElGamal public key and $(c_1=g_2, c_2)$ as ciphertext.

Then the oracle outputs

$$c_2/m = g^{(\log_g g_1 \log_g g_2)} \pmod p$$

which is a solution for the CDH instance.