

# Cryptography CS 555

## Lecture 13



Department of Computer Sciences  
Purdue University

# Review of RSA

**Public key: (e, n)**

**Secret key: d**

where  $n=pq$  and  $ed \equiv 1 \pmod{\Phi(n)}$

Encrypting M:  $M^e \pmod n$

Decrypting C:  $C^d \pmod n$

- Three equivalent approaches to do key discovery:
  1. Factor  $n = pq$
  2. Determine  $\Phi(n)$
  3. Find the private key  $d$  directly

# Review of Attacks on RSA

- **Common modulus attack**
- **Small encryption exponent  $e$**
- **Small decryption exponent  $d$**
- **Multiplicative property**

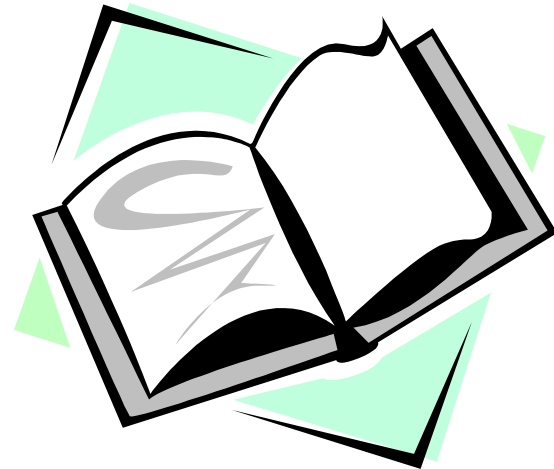
# Lecture Outline

- Rabin
- ElGamal
- Probabilistic encryption schemes



# Recommended Reading

- Stinson: Chapter 5.8  
(for Rabin)
- Stinson: Chapter 6.1  
(ElGamal)
- HAC: Chapter 8.7  
(probabilistic schemes)



# Quadratic Residues

- $a$  is a quadratic residue modulo  $p$  if  $\exists b \in \mathbb{Z}_p^*$  such that  $b^2 \equiv a \pmod{p}$ , otherwise  $a$  is a nonquadratic residue
- $\underline{Q}_p$  is the set of all quadratic residues
- $\overline{Q}_p$  is the set of all nonquadratic residues
- If  $p$  is prime there are  $(p-1)/2$  quadratic residues in  $\mathbb{Z}_p^*$ ,  $|Q_p| = (p-1)/2$

# Euler's Criterion

- Theorem: If  $r^{(p-1)/2} \equiv 1 \pmod{p}$  then  $r$  is a quadratic residue (if  $-1$  then  $r$  is a nonquadratic residue)
- Proof. if  $r = y^2$ , then  $r^{(p-1)/2} = y^{(p-1)} = 1 \pmod{p}$
- If  $r^{(p-1)/2} = 1$ , let  $r = a^i$ , where  $a$  is a generator of the group  $Z_p^*$ . Then  $a^{i(p-1)/2} = 1 \pmod{p}$ . Since  $a$  is a generator,  $p-1 \mid i(p-1)/2$ , thus  $i$  must be even. Therefore,  $r$  is QR.

# Legendre Symbol

- Let  $p$  be an odd prime and  $a$  an integer. The Legendre symbol is defined

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p \mid a \\ 1, & \text{if } a \in Q_p \\ -1, & \text{if } a \in \overline{Q}_p \end{cases}$$

# Jacobi Symbol

- let  $n \geq 3$  be odd with prime factorization

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

- the Jacobi symbol is defined to be

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

# Rabin Public Key Encryption

- The first example of a provably-secure public key encryption scheme.
- Recovering the plaintext is computationally equivalent to factoring.

## Key Generation

select  $p$  and  $q$  primes s.t.  $p \equiv 3 \pmod{4}$  and  $q \equiv 3 \pmod{4}$

$n = pq$

Public key:  $n$

Private key:  $(p, q)$

# Rabin Cryptosystem

## Encryption

$$c = m^2 \pmod{n}$$

## Decryption

$$m = \sqrt{c} \pmod{n}$$

Decryption is not an injection:

Find the four square roots

$m_1$  ,  $m_2$  ,  $m_3$ , and  $m_4$  of  $c \pmod{n}$

The message sent was either  $m_1$  ,  $m_2$  ,  $m_3$ , or  $m_4$

# Finding Square Root

- Given  $c$ , finds out  $x$  such that  $x^2 \equiv c \pmod n$
- $(c^{(p+1)/4})^2 \equiv c^{(p+1)/2} \equiv c^{(p-1)/2} \cdot c \equiv c \pmod p$ 
  - $x \pmod p$  should be  $c^{(p+1)/4}$  or  $-c^{(p+1)/4}$
- Similarly,  $x \pmod q$  should be  $c^{(q+1)/4}$  or  $-c^{(q+1)/4}$
- Use Chinese Remainder Theorem to solve it
  - Find integers  $a, b$  such that  $ap + bq = 1$ .
  - Compute  $r = c^{(p+1)/4} \pmod p$  and  $s = c^{(q+1)/4} \pmod q$ .
  - Compute  $m = (aps + bqr) \pmod n$ .
  - Compute  $t = (aps - bqr) \pmod n$ .
  - The four square roots of  $c$  modulo  $n$  are  $m, -m \pmod n, t,$  and  $-t \pmod n$ .
- For  $p \equiv 1 \pmod 4$ , there is no known PTIME deterministic algorithm to compute square roots modulo  $p$

# Security of Rabin

- Provably secure against **passive adversary**, relying on the difficulty of factoring large composites
- Obtaining plaintext from the ciphertext is equivalent to the modulo square root problem
- Modulo square root problem is equivalent to prime factoring
- Susceptible to chosen ciphertext attack similar to RSA
- Many RSA attacks can be also applicable to Rabin: solution similar to OAEP

# ElGamal

- Published in 1985 by ElGamal
- Its security based on the intractability of the discrete logarithm problem
- Message expansion: the ciphertext is twice as big as the original message
- Uses randomization, each message has  $p-1$  possible different encryptions

# Discrete Logarithm Problem (DLP)

- Given a multiplicative group  $(G, *)$ , an element  $g$  in  $G$  having order  $n$  and an element  $y$  in the group generated by  $g$ , denoted  $\langle g \rangle$
- Find the unique integer  $x$  such that

$$g^x \bmod n = y$$

- $x$  is the discrete logarithm

# Diffie-Hellman Key Exchange

- A and B wishes to establish a shared secret key so that no eavesdropper can compute the key:
- A and B shares public parameters a group  $Z_p$  and a generator  $g$ 
  - A randomly chooses  $x$  and send  $g^x \bmod p$  to B
  - B randomly chooses  $y$  and send  $g^y \bmod p$  to A
  - Both A and B can compute  $g^{xy} \bmod p$
  - It is (believed to be) infeasible for an eavesdropper to compute  $g^{xy} \bmod p$

# El Gamal

## Key Generation

- Generate a large random prime  $p$  such that DLP is infeasible in  $Z_p$  and a generator  $g$  of the multiplicative group  $Z_p$  of the integers modulo  $p$
- Select a random integer  $a$ ,  $1 \leq a \leq p-2$ , and compute
$$g^a \bmod p$$
- Public key is  $(p; g ; g^a)$
- Private key is  $a$ .

# El Gamal (cont.)

## Encryption:

Message  $M$  into ciphertext  $C$

Select a random integer  $k$ ,  $1 \leq k \leq p-2$ .

Compute  $\gamma = \alpha^k \bmod p$  and  $\delta = m (g^a)^k \bmod p$ .

Ciphertext  $C = (\gamma, \delta)$

## Decryption:

Compute  $\gamma^{-a}$  as follows:  $\gamma^{p-1-a} \bmod p = \gamma^{-a} \bmod p$

$m = \gamma^{-a} \delta \bmod p$

WHY DECRYPTION WORKS?

$$\gamma^{-a} \delta \bmod p \equiv g^{-ka} m \cdot (g^a)^k \bmod p \equiv m \bmod p$$

# Parameters Size

- All parties could use the same modulus  $p$  and generator  $g$
- Different encryptions should use different  $k$
- Prime  $p$  should be chosen as 1024 bits to ensure that DLP is infeasible, while  $k$  should be 160 bits
- Several algorithms to compute discrete logarithms.
- DLP can be also defined in elliptic curves cryptography. DLP is more difficult in such a setup.

# Security of ElGamal

- ElGamal is not semantically secure.
- WHY? An attacker can learn information about the plaintext without decrypting: given two encryptions, can say which plaintext was a quadratic residue and which one was not.
- Semantically secure cryptosystems: an adversary can not, in polynomial time, distinguish ciphertexts.

# Semantically Secure ElGamal

- Choose  $p$  such that  $p = 2q + 1$ , where  $q$  is also prime
- Then define ElGamal in  $Q_p$ , the subgroup of quadratic residues modulo  $p$ , this subgroup is a cyclic subgroup of  $Z_p$  having order  $q$
- Equivalent with restricting the message  $m$ ,  $\alpha^a$  and  $y_1 = \alpha^k \pmod p$  to be quadratic residues

# Computational Diffie-Hellman (CDH)

- Computational Diffie-Hellman (CDH)
  - Given a multiplicative group  $(G, *)$ , an element  $g \in G$  having order  $n$ , given  $g^x$  and  $g^y$ , find  $g^{xy}$
- Decision Diffie-Hellman (DDH)
  - Given a multiplicative group  $(G, *)$ , an element  $g \in G$  having order  $n$ , given  $g^x$ ,  $g^y$ , and  $g^z$ , determine if  $g^x g^y \equiv g^z \pmod{n}$

# ElGamal and DH Problems

- Semantic security of ElGamal is equivalent to the infeasibility of Decision Diffie-Hellman
- ElGamal decryption (without knowing the public key) is equivalent to solving Computational Diffie-Hellman

# CDH and ElGamal

*Proove that any algorithm that solves CDH can be used to decrypt ElGamal ciphertxts*

“=>” Assume that algorithm OracleCHD solves CHD and let  $(y_1, y_2)$  be an ElGamal encryption

public key  $(\alpha, \beta = \alpha^a)$

$$y_1 = \alpha^k \text{ mod } p, y_2 = m (\alpha^a)^k \text{ mod } p$$

$\delta = \text{OracleCDH}(\alpha, \beta, y_1)$  and

$$x = y_2 \delta^{-1}$$

then  $x$  is the decryption of  $(y_1, y_2)$

# CDH and ElGamal (cont.)

“ $\leq$ ” Assume that algorithm OracleElGamalDec decrypts ElGamal ciphertexts

Define  $y_1 = \gamma$  and

let  $y_2$  in  $\langle \alpha \rangle$  be chosen randomly

$x = \text{OracleElGamalDec}(\alpha, \beta, (y_1, y_2))$  and

$$\delta = y_2 x^{-1}$$

then  $\delta$  is the solution of CDH

# Probabilistic Public-Key Encryption

- Deterministic Encryption Systems
  - The same message gets encrypted to the same ciphertext
  - In RSA, adversary can get one bit info from the encrypted message
  - Solution to these problems: use random padding
  - The resulting schemes are generally not provably secure
- Probabilistic Public Key Encryption Schemes (PPKE) uses randomness to attain provable security

# Quadratic Residuosity Problem

- Given
  - an odd composite integer  $n$  and
  - a number  $a$  such that the Jacobi symbol of  $a$  is 1,
- Decide if  $a$  is a quadratic residue or not
- Note: if  $n$  is prime then it is easy to say if  $a$  is a quadratic residue or not, we can calculate the Legendre symbol, if 1 then  $a$  is a quadratic residue

# Goldwasser-Micali

- Semantically secure assuming the intractability of the quadratic residuosity problem
- Message expansion by a factor of  $\log n$ .

## Key Generation

- Select two large random and distinct primes  $p$  and  $q$ , of about the same size.
- Compute  $n = pq$ .
- Select  $y \in \mathbb{Z}_n$  such that  $y$  is a quadratic non-residue modulo  $n$  and the Jacobi symbol  $(y/n) = 1$
- Public key is  $(n, y)$ ;
- Private key is the pair  $(p, q)$

# Goldwasser-Micali

## Encryption

- Represent the message  $m$  as a binary string
- $M = m_1, m_2, \dots, m_t$  of length  $t$ .
- For  $i$  from 1 to  $t$  do:
  - i. Select  $x \in \mathbb{Z}_n^*$  at random
  - ii. If  $m_i = 1$ , then  $c_i \leftarrow yx^2 \bmod n$ ;
  - else  $c_i \leftarrow x^2 \bmod n$ .
- $C = (c_1, c_2, \dots, c_t)$

# Goldwasser-Micali

## Decryption

- For  $i$  from 1 to  $t$  do:
  - i. Compute the Legendre symbol  $e_i = (c_i/p)$
  - ii. If  $e_i = 1$  then set  $m_i \leftarrow 0$ ; otherwise  $m_i \leftarrow -1$
- The decrypted message is  $M = m_1 m_2 \dots m_t$

# Blum-Goldwasser

- One of the most efficient PPKE scheme
- Comparable to RSA in terms of speed and message expansion
- Semantically secure assuming intractability of integer factorization

# Blum-Goldwasser

## Key Generation

select  $p$  and  $q$  primes s.t.

$p \equiv 3 \pmod{4}$  and  $q \equiv 3 \pmod{4}$

Compute  $n = pq$ .

Compute integers  $a$  and  $b$  such that  $ap + bq = 1$ .

Public key:  $n$

Private key:  $(p, q, a, b)$

# Blum-Goldwasser

## Encryption

$k = \lfloor \lg n \rfloor$  and  $h = \lfloor \lg k \rfloor$ .

Represent message  $m$  as a string  $m = m_1 m_2 \dots m_t$  of length  $t$ , where each  $m_i$  is a binary string of length  $h$

Select as a seed  $x_0$ , a random quadratic residue modulo  $n$ .

For  $i$  from 1 to  $t$  do the following:

- i. Compute  $x_i = x_{i-1}^2 \bmod n$
- ii. Let  $p_i$  be the  $h$  least significant bits of  $x_i$
- iii. Compute  $c_i = p_i \oplus m_i$

Compute  $x_{t+1} = x_t^2 \bmod n$

Ciphertext  $c = (c_1, c_2, \dots, c_t, x_{t+1})$

# Blum-Goldwasser

## Decryption

Compute  $d_1 = ((p + 1)/4)^{t+1} \bmod (p - 1)$

Compute  $d_2 = ((q + 1)/4)^{t+1} \bmod (q - 1)$ .

Compute  $u = x_{t+1}^{d_1} \bmod p$

Compute  $v = x_{t+1}^{d_2} \bmod q$

Compute  $x_0 = vap + ubq \bmod n$ .

For  $i$  from 1 to  $t$  do the following:

i. Compute  $x_i = x_{i-1}^2 \bmod n$

ii. Let  $p_i$  be the  $h$  least significant bits of  $x_i$

iii. Compute  $m_i = p_i \oplus c_i$

# Summary

- ElGamal based on DLP, message expansion
- ElGamal in  $Q_p$  is semantically secure
- Probabilistic schemes use randomness to attain provable security
- Probabilistic schemes also have message expansion



# Next Lecture..

- Digital Signatures
  - RSA
  - ElGamal
  - DSA
  - Schnorr

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.