# Introduction to Cryptography
# Lecture 5

## Public key encryption
## El Gamal, RSA cryptosystems

Benny Pinkas

# Number Theory

- Lagrange's Theorem: $\forall a$ in a finite group $G$, $a^{|G|}=1$.
- Euler's phi function (aka, Euler's totient function),
  - $\Phi(n)$ = number of elements in $Z^*_n$ (i.e. $|\{x \mid gcd(x,n)=1,\ 1 \le x \le n\}|$
  - $\Phi(p) = p-1$ for a prime $p$.
  - $N=\prod_{i=1..k} p_i^{e(i)} \Rightarrow \Phi(n) = n \cdot \prod_{i=1..k} (1-1/p_i)$
  - $\Phi(p^2) = p(p-1)$ for a prime $p$.
  - $N=p \cdot q \Rightarrow \Phi(n) =(p-1)(q-1)$

- Corollary: $\forall a \in Z_n^*$ it holds that $a^{\Phi(n)} =1 \bmod n$
  - For $Z_p^*$ (prime $p$), $a^{p-1} =1 \bmod p$ (Fermat's theorem).
  - For $Z_n^*$ ($n=p \cdot q$), $a^{(p-1)(q-1)} =1 \bmod n$

# Prime Number Theorem

- Prime number theorem: $\#\{\text{primes} \le x\} \approx x / \ln x$ as $x \to \infty$
- How can we find a k-bit prime?
  - Choose x in $\{2^k, \ldots, 2^{k+1}-1\}$
  - Test if $x$ is prime
- The probability of success is $\approx 1/\ln(2^k) = 1/k$.
- The expected number of trials is $O(k)$.
- How can we find a generator of $Z_p^*$?
- Can check whether $\forall\, 1 \le i \le p-2 \quad a^i \ne 1$ ☹
- Easy if we know the factorization of $(p-1)$
  - For all $a \in Z_p^*$, the order of $a$ divides $(p-1)$
  - For every integer divisor $b$ of $(p-1)$, check if $a^b = 1 \bmod p$.
  - $a$ is a generator iff $ord(a) = p-1$.

# Quadratic Residues

- The square root of $x \in Z_p^*$ is $y \in Z_p^*$ s.t. $y^2 = x$ mod $p$.
- Examples: sqrt(2) mod 7 = 3, sqrt(3) mod 7 doesn't exist.
- How many square roots does $x \in Z_p^*$ have?
  - If $x = a^2 = b^2$ mod $p$ then $(a-b)(a+b) = 0$ mod $p$. Therefore either $a = b$ or $a = -b$ modulo $p$.
  - Therefore $x$ has either $2$ or $0$ square roots, and is denoted as a Quadratic Residue (QR) or Non Quadratic Residue (NQR), respectively.
- $a^{(p-1)/2}$ is either 1 or -1 in $Z_p^*$. (indeed, $(a^{(p-1)/2})^2$ is always $1$)
- Euler's theorem: $x \in Z_p^*$ is a QR iff $x^{(p-1)/2} = 1$ mod $p$.
- *Legendre's symbol:*

$$\left(\frac{x}{p}\right) = \begin{cases} 1 & x \text{ is a QR in } Z_p^* \\ -1 & x \text{ is an NQR in } Z_p^* \\ 0 & x = 0 \bmod p \end{cases}$$

- Can be efficiently computed as $x^{(p-1)/2}$ mod $p$.

# Hard problems in $Z_p^*$

- The following problems are believed to be hard
- Discrete logarithm: let $g$ be a generator of $Z_p^*$. Given a random $x \in Z_p^*$ find an $r$ such that $x=g^r \bmod p$.
- The Diffie-Hellman problem: Given random $x,y \in Z_p^*$, such that $x=g^a$ and $y=g^b$, *find* $z=g^{a \cdot b}$.
- The Decisional Diffie-Hellman (DDH) problem: Given random $x,y \in Z_p^*$, such that $x=g^a$ and $y=g^b$; *and a value* $z$ which is promised to be either $g^{a \cdot b}$ *or* $g^c$ (for a random c), tell which is the case.
- DL > DH > DDH

# Does the DDH assumption hold in $Z_p^*$?

- The DDH assumption does not hold in $Z_p^*$
  - Assume that the Legendre symbol of $x=g^a$ and $y=g^b$ is 1
  - I.e., both are QRs in $Z_p^*$. ($a,b$ are even. $x^{(p-1)/2}= y^{(p-1)/2}=1$.)
  - Then the Legendre symbol of $g^{ab}$ is always $1$, whereas the symbol of a random $g^c$ is $1$ with probability $1/2$.
- Solution: (work in a subgroup of prime order)
  - Set $p=2q+1$, where $q$ is prime.
  - $\Phi(Z_p^*) = p-1 = 2q$. Therefore has a subgroup $H$ of *prime* order $q$.
  - Let $g$ be a generator of $H$.
  - The DDH assumption is believed to hold in $H$. (The Legendre symbol is always 1.)

# The Diffie-Hellman Key Exchange Protocol

- Public parameters: a group $Z_p^*$ (where $|p|= 768$ or $1024$, $p=2q+1$), and a generator $g$ of $H \subset Z_p^*$ of order $q$.

- Alice:
  - picks a random $a \in [1,q]$.
  - Sends $g^a$ mod $p$ to Bob.
  - Computes $k=(g^b)^a$ mod $p$

- Bob:
  - picks a random $b \in [1,q]$.
  - Sends $g^b$ mod $p$ to Bob.
  - Computes $k=(g^a)^b$ mod $p$

- $K = g^{ab}$ is used as a shared key between Alice and Bob.
  - DDH assumption $\Rightarrow$ $K$ is indistinguishable from a random key
  - $K$ is a master key which is used to encrypt session keys. Session keys are used to encrypt traffic with a symmetric cryptosystem

# An active attack against the Diffie-Hellman Key Exchange Protocol

- An active adversary Eve.
- Can read and change the communication between Alice and Bob.
- …As if Alice and Bob communicate via Eve.

# An active attack against the Diffie-Hellman Key Exchange Protocol

- Alice:
  - picks a random $a \in [1,q]$.
  - Sends $g^a$ mod $p$ *to Bob.*
- Bob:

Eve changes $g^a$ to $g^c$ →

  - picks a random $b \in [1,q]$.
  - Sends $g^b$ mod $p$ to Alice.

Eve changes $g^b$ to $g^d$ ←

  - *Computes* $k=(g^d)^a$ mod $p$
  - *Computes* $k=(g^c)^b$ mod $p$

Keys:

| Alice | Eve | Bob |
|---|---|---|
| $g^{ad}$ | $g^{ad}$, $g^{bc}$ | $g^{bc}$ |

  - Solution: ?

# Public key encryption

- Alice publishes a public key $PK_{Alice}$.
- Alice has a secret key $SK_{Alice}$.
- Anyone knowing $PK_{Alice}$ can encrypt messages using it. (No need for an interactive key agreement protocol.)
- Message decryption is possible only if $SK_{Alice}$ is known.

- Easier key management:
  - *n* users need *n* keys rather than $O(n^2)$
- Secure as long as we can trust the association of keys with users.

# The El Gamal public key encryption system

- (Find the similarity with Diffie-Hellman key exchange)
- Public information (can be common to different public keys):
  - A prime $p=2q+1$, and a generator $g$ of $H \subset Z_p^*$ of order $q$.
- Private key: $0 < a < q$.
- Public key: $h=g^a \bmod p$.

- Encryption of message $m \in H \subset Z_p^*$
  - Pick a random $0 < r < q$.
  - The ciphertext is $(g^r, h^r \cdot m)$.

  } Using public key alone

- Decryption of $(s,t)$
  - Compute $t / s^a$ $(m= h^r \cdot m / (g^r)^a)$

  } Using private key

# El Gamal and Diffie-Hellman

- ElGamal encryption is similar to DH key exchange
  - DH key exchange: $g^a$, $g^b$, cannot distinguish $g^{ab}$ *from random*
  - El Gamal:
    - A fixed public key $g^a$.
    - Sender picks a random $g^r$.

    } Known to the adversary
    - Encrypt message using $g^{ar}$. } Used as a key

- El Gamal is like DH where
  - The same $g^a$ ($g^r$) is used for all communication
  - There is no need to explicitly send this $g^a$

# The El Gamal public key encryption system

- Setting the public information
- *A large prime p, and a generator g of* $H \subset Z_p^*$ of order *q.*
  - $|p|$ = 756 or 1024 bits.
  - *p-1* must have a large prime factor (e.g. *p=2q+1*)
    - Otherwise it's easy to solve discrete logs in $Z_p^*$ (relevant also to DH key agreement)
    - Needed for the DDH assumption to hold (Legendre's symbol)
  - *g* must be a generator of a large subgroup of $Z_p^*$.

- Encoding the message:
  - *m* must be in the subgroup generated by *g.*
  - Alternatively, encrypt m using $(g^r, H(h^r) \oplus m)$. *Decryption is done by computing* $H((g^r)^a)$. (*H* is a hash function that preserves the pseudo-randomness of $h^r$.)

# The El Gamal public key encryption system

- Overhead:
  - Encryption: two exponentiations; preprocessing possible.
  - Decryption: one exponentiation.
  - 2× message expansion: $m \in Z_p^* \Rightarrow (g^r, h^r \cdot m)$.

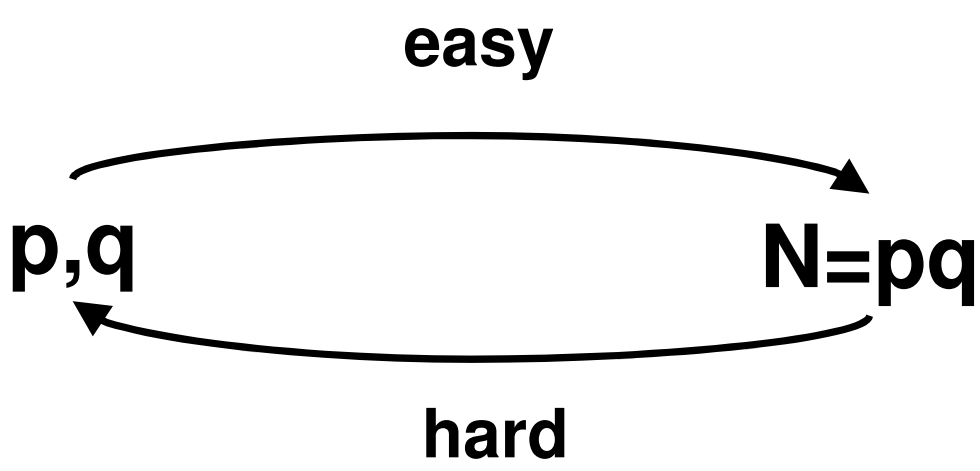
- Randomized encryption
  - Must use fresh randomness $r$ for every message.
  - (Good) Semantic security: two different encryptions of the same message are different.

# Homomorphic property

- Insecurity against chosen ciphertext attacks:
  - Attacker wants to decrypt $(s,t) = (g^r, h^r \cdot m)$.
  - Chooses random r', computes $(s',t')=(s, t \cdot r') = (g^r, h^r \cdot (m \cdot r'))$.
  - Asks for a decryption of $(s',t')$. Receives $m \cdot r'$.

- Homomorphic property:
  - Given encryptions of $x,y$, it's easy to generate an encryption of $x \cdot y$.
    - $(g^r, h^r \cdot x) \times (g^{r'}, h^{r'} \cdot y) \rightarrow (g^{r''}, h^{r''} \cdot x \cdot y)$

# Integer Multiplication & Factoring as a One Way Function.




Can a public key system be based on this observation ?????

# Excerpts from RSA paper (CACM, 1978)

The era of "electronic mail" may soon be upon us; we must ensure that two important properties of the current "paper mail" system are preserved: (a) messages are *private*, and (b) messages can be *signed*. We demonstrate in this paper how to build these capabilities into an electronic mail system.

At the heart of our proposal is a new encryption method. This method provides an implementation of a "public-key cryptosystem," an elegant concept invented by Diffie and Hellman. Their article motivated our research, since they presented the concept but not any practical implementation of such system.

# The Multiplicative Group $Z_{pq}^*$

- $p$ and $q$ denote two large primes (e.g. 512 bits long).
- Denote their product as $N = pq$.
- The multiplicative group $Z_N^* = Z_{pq}^*$ contains all integers in the range $[1, pq-1]$ that are relatively prime to both $p$ and $q$.

- The size of the group is
  - $\varphi(n) = \varphi(pq) = (p-1)(q-1) = N - (p+q) + 1$
- For every $x \in Z_N^*$, $x^{(p-1)(q-1)} = 1 \bmod N$.

# Exponentiation in $Z_N^*$

- Motivation: use exponentiation for encryption.

- Let $e$ be an integer, $1 < e < \varphi(N) = (p-1)(q-1)$.
  - Question: When is exponentiation to the $e^{th}$ power, $x \dashrightarrow x^e$, a one-to-one operation in $Z_N^*$ ?

- Claim: If $e$ is relatively prime to $(p-1)(q-1)$ then $x \dashrightarrow x^e$ is a one-to-one operation in $Z_N^*$.
- Constructive proof:
  - Since $\gcd(e, (p-1)(q-1)) = 1$, $e$ has a multiplicative inverse mod $(p-1)(q-1)$.
  - Denote it by $d$, then $ed = 1 + c(p-1)(q-1) = 1 + c\varphi(N)$.
  - Let $y = x^e$, then $y^d = (x^e)^d = x^{1+c\varphi(N)} = x$.
  - I.e., $y \dashrightarrow y^d$ is the inverse of $x \dashrightarrow x^e$.

# RSA Public Key Cryptosystem

- Public key*:*
  - $N=pq$ the product of two primes
  - $e$ such that $\gcd(e,\phi(N))=1$ *(are these hard to find?)*
- Private key:
  - $d$ such that $de\equiv 1 \bmod \phi(N)$

- Encryption of $M\in Z_N^*$
  - $C=E(M)=M^e \bmod N$
- Decryption of $C\in Z_N^*$
  - $M=D(C)=C^d \bmod N$ *(why does it work?)*

# Constructing an instance of RSA PKC

- Alice
  - picks at random two large primes, $p$ and $q$.
  - picks at random a large $d$ that is relatively prime to (p-1)(q-1) ( $\gcd(d,\phi(N))=1$ ).
  - Alice computes $e$ such that $de \equiv 1 \bmod \phi(N)$

- Let $N=pq$ be the product of $p$ and $q$.
- Alice publishes the public key $(N,e)$.
- Alice keeps the private key $d$, as well as the primes $p$, $q$ and the number $\phi(N)$, in a safe place.

# Properties of RSA

- Deterministic encryption. In textbook RSA:
  - $M$ is always encrypted as $M^e$
  - The ciphertext is as long as the domain of $M$
- The public exponent $e$ may be small. It's common to choose its value to be either 3 or $2^{16}+1$. The private key $d$ must be long.
  - Each encryption involves several modular multiplications. Decryption is longer.
- Chosen ciphertext attack: (homomorphic property)
  - Given a ciphertext $C=M^e$, choose a random $R$ and generate $C'=CR^e$ (an encryption of $M \cdot R$). A decryption of $C'$ reveals $M$.