Cryptography and Security Summary

Basic Definition: What is cryptography? What is cryptanalysis? What is cryptology? What is information security? What is cyber security? Three important concerns of information security (confidentiality, integrity, availability). Security attack vs security threat. Types of attacks: passive and active. Security threats in networked world: interception, modification, fabrication, non-repudiation, interruption.

Sets:

A set is a collection of objects. The objects are referred to as elements of the set.

Example:

$X = \{a, b, c\}$ is a set with three elements a, b and c.

Name	Set	Symbol Used
Natural Numbers	$\{0, 1, 2, 3, \cdots\}$	N
Integers	$\{\cdots, -2, -1, 0, +1, +2, \cdots\}$	Z
Positive Integers	$\{1, 2, 3, \cdots\}$	$Z+$
Negative Integers	$\{\cdots, -2, -1\}$	$Z-$

Function: What is a function? What is domain? What is codomain?

A function is defined by a triplet $< X, Y, f >$, where

- X: a set called domain;
- Y: a set called range or codomain and
- f: a rule which assigns to each element in X precisely one element in Y. It is denoted by $f : X \rightarrow Y$

Image : If $x \in X$, the image of x in Y is an element $y \in Y$ such that $y = f(x)$.

Pre-image : If $y \in Y$, then a Pre-image of y in X is an element $x \in X$ such that $f(x) = y$.

Image of a function f **(**$Im(f)$: A set of all elements in Y which have at least one Pre-image.

One-on-one Function:

A function is one-to-one (injective) if each element in the codomain Y is the image of **at most** one element in the domian X. In other words, each element in x in X is related to different y in X, never two different elements in X map to a same element in Y. We can say that $|X| \leq |Y|$. An alternate definition would be, a $f : X \to Y$ is one-to-one (injective), provided

$$f(x_1) = f(x_2) \text{ implies } x_1 = x_2.$$

Examples: Let $X = Y = \mathbf{Z}_4$, Then $f : X \to Y$ given by $f(x) = 3 * x$ is a one-to-one function. However $f(x) = x^2$ is a not a one-to-one function.

Onto Function:

A function is Onto (surjective) if each element in the codomain Y is the image of **at least** one element in the domian X.
A function $f : X \to Y$ is onto if $Im(f) = Y$
We can say that, if f is onto then $|Y| \leq |X|$.
Example: Let $X = Y = \mathbf{Z}_5$, Then $f : X \to Y$ given by $f(x) = x^2$ is a onto function.

Bijection:

Bijection: A function which is both one-to-one and onto.
In this case, we have $|X| \leq |Y|$ and $|Y| \leq |X|$. This implies $|X| = |Y|$.
If $f : X \to Y$ is one-to-one then $f : X \to Im(f)$ is a bijection.
If $f : X \to Y$ is onto and X and Y are finite sets of the same size then f is a bijection.

Divisibility:

An integer "a" is said to be **divisible** by a positive integer "b", and this is written as $b|a$, if $a = b\ c$ for a third integer "c" and $c \neq 0$. (The above statement is also same as "b" divides "a".)
In the following statements, a, b, c are integers.

1. $a|a$,
2. $a|b$ and $b|c$ implies $a|c$,
3. $a|b$ and $b|a$ implies $a = \pm b$,
4. $a|b$ and $a|c$ implies $a|(b\ x + c\ y)$ for all integers x and y,
5. $a|b$ implies $ca\ |cb$, for any c.

Quotient and Remainer:

Let a be any integer b a positive integer which is not zero, then are unique integers q (quotient) and r (remainder) such that

$$a = qb + r, 0 \leq r < b.$$

Prime Number:

Definition

A number is said to be **a prime number** *if* $p > 1$ *and* p *has no positive divisors except* 1 *and* p.

Greatest Common Divisor:

Definition

If d *divides two integers* m *and* n, *then* d *is called a common divisor. The greatest of common divisors of the integers is the GCD of* m *and* n.

Definition

Numbers m *and* n *are said to be relatively prime if the GCD of* m *and* n *is* 1.

Euclidean Algorithm:

Fact

Let $a > b > 0$. *Then*

$$gcd(a, b) = gcd(b, (a \ mod \ b)).$$

Modular Arithmetic:

- Congruent:

  We say "a" is congruent to "b", modulo n and write

  $$a \equiv b \ (mod \ \ n),$$

- Addition and Multiplication:

  $$X \oplus_n y = (x + y) \ mod \ n.$$

  $$X \otimes_n y = (xy) \ mod \ n$$

- Modular Inverse:

Definition

Let $x \in Z_n$, *if there is an integer* y *such that*

$$X \otimes_n y = 1,$$

then we say y *is the multiplicative inverse of* x. *It is denoted by* $y = x^{-1}$ *usually.*

- Factorization Theorem:

Fact

Every natural number $n > 1$ has a unique prime factorization or prime power factorization.

$$n = \Pi_{i=1}^{\tau} p_i^{a_i},$$

where τ is a positive number.

Extended Euclidean Algorithm:

Fact

For any integers a and b, there exist integers x and y such that

$$gcd[a, b] := ax + by.$$

The requirement is a should be relatively prime to b, i.e., gcd(a, b) = 1.

Consider $gcd(13, 25)$:

$$\begin{array}{rcll} 25 & = & 1 \times 13 + 12 & gcd(13, 12) \quad (A) \\ 13 & = & 1 \times 12 + 1 & gcd(12, 1) \quad (B) \\ 12 & = & 12 \times 1 + 0 & gcd(1, 0) \end{array}$$

Table: Determine $gcd(13, 25)$

$$\begin{array}{rl} 1 = 13 - 1 \times 12 & From(B) \\ 1 = 13 - 1 \times (25 - 1 \times 13) & From(A) \\ 1 = 2 \times 13 - 1 \times 25 & \\ 1 = 2 \times 13 + (-1) \times 25 & Simplification \end{array}$$

It is easy to see now, 2 is inverse of 13 mod 25.

Euler Phi Function:

Definition

Euler phi function(or Euler totient function): For $n \geq 1$, let $\phi(n)$ denote the number of integers less than n but are relatively prime to n.

Properties:

- n is a prime number:

Fact

$\phi(p) = p - 1$, *for any prime* p.

- n is the power of a prime number:

Fact

$$\phi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1),$$

for any prime p *and any integer* $a \geq 1$.

- n is the multiplication of two prime numbers:

Fact

$\phi(pq) = (p-1)(q-1)$, *for any pair of primes* p *and* q.

- n is the multiplication of two relatively prime numbers:

Fact

If a *and* b *are relatively prime numbers (* $gcd(a, b) = 1$*), then,*

$$\phi(ab) = \phi(a)\phi(b).$$

Set of Residues:

Definition

Reduced set of residues mod n*: For* $n \geq 1$*, the reduced set of residues,* $R(n)$ *is defined as set of residues modulo* n *which are relatively prime to* n.

Symmetry Key Encryption: A same key used for both encryption and decryption.

Kerckhoffs's Principle: All algorithm details are public and security should be obtained by using only secrecy of key.

Stream Ciphers and Block Cipher:

- Stream ciphers: plaintexts are streamed to cipher producing stream of ciphertexts element by element.
- Block Ciphers: plaintext is divided into blocks of data, cipher process one block at a time

Types of Cryptanalytic Attacks: ciphertext only, known plaintext, chosen plaintext, chosen ciphertext, chosen text.

Two Aspects of Security: unconditional security, computational security.

Classical Cipher:

- Substitution cipher:

– Here plaintext symbols are substituted or replaced with other symbols using an unknown key.
– The substitutions can be performed as sequence of symbols or symbol by symbol.

(i) Caesar cipher:

Encryption: $E(k,p) = c = p + k \bmod 26$

Decryption: $D(k,p) = c - k \bmod 26$

Complexity: 26; Cryptanalysis: brute force.

(ii) Affine cipher:

Encryption: $E(k,p) = c = ap + b \bmod 26$

Decryption: $D(k,p) = \text{Inverse}(a)(c - b) \bmod 26$

Complexity: $26*\varphi(26)$; Cryptanalysis: brute force.

(iii) Monoalphabetic cipher:

Key is a permutation of plaintext space character.

Complexity: 26!; Cryptanalysis: statistics approach for language frequency.

- Transposition cipher:

– Here plaintexts are organized as a sequence of plaintext blocks and symbol positions in each block are permuted or transposed using a key. The same permutation is used for every block

(i) Rail fence cipher:

Plaintext: meetmeafterthetogaparty

Write as:
```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

Ciphertext: mematrhtgpryetefeteoaat

(ii) Row transposition cipher:

```
Key:          4 3 1 2 5 6 7
Plaintext:    a t t a c k p
              o s t p o n e
              d u n t i l t
              w o a m x y z
Ciphertext:   TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Complex Cipher:

- Polyalphabetic cipher:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

- Vigenere cipher:

$P = p_1\ p_2 \cdots p_d\ p_{d+1}\ p_{d+2} \cdots p_{2d....}$

$C = c_1\ c_2 \ldots c_d\ c_{d+1}\ c_{d+2} \ldots c_{2d....}$

Encryption: $E(K,P) = C$, where $c_i = p_i + k_i \bmod 26$

Decrypton: $D(K,C) = P$, where $p_i = c_i - k_i \bmod 26$

Modern Cipher:

- Vernam cipher(One-Time-Pad): Perfect secrecy.

$P = p_1\ p_2 \ldots p_d\ p_{d+1}\ p_{d+2} \ldots p_{2d\ldots}$

$C = c_1\ c_2 \ldots c_d\ c_{d+1}\ c_{d+2} \ldots c_{2d\ldots}$

Encryption: $E(K,P) = C$, where $c_i = p_i + k_i \bmod 26$

Decrypton: $D(K,C) = P$, where $p_i = c_i - k_i \bmod 26$

But the size of key equals the size of message, and the each key is chosen randomly.

$C_1 = M_1 \oplus K$; $C_2 = M_2 \oplus K$; then

$C_1 \oplus C_2 = M_1 \oplus M_2 \oplus K \oplus K = M_1 \oplus M_2$.

Even though $M_1 \oplus M_2$ may not direct meaning, it still leaks information about both M_1 and M_2. Also, in a cryptanalysis setting if one of the messages M_1 or M_2 is available to the adversary, then he/she can get the other.

This attack implies that you need a new key for every message.

- Product cipher:

  (i) Confusion and Diffusion:

  Diffusion dissipates statistical structure of plaintext over bulk of ciphertext.
  Confusion makes relationship between ciphertext and key as complex as possible.

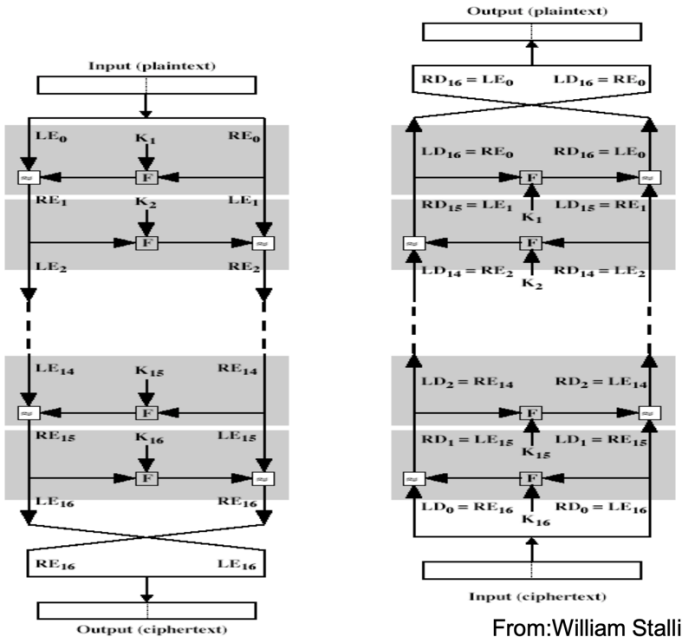
  (ii) Substitution-Permutation Cipher:

  A **substitution-permutation cipher** is a product cipher made up of number of stages each involving substitution and permutation. The operations of substitution and permutation are responsible for effecting the confusion and diffusion respectively.

  (iii) Iterated Cipher:

  An **iterated block cipher** is a block cipher involving sequential repetition of an iterated function called a round function.

  (iv) Fiestel Cipher:

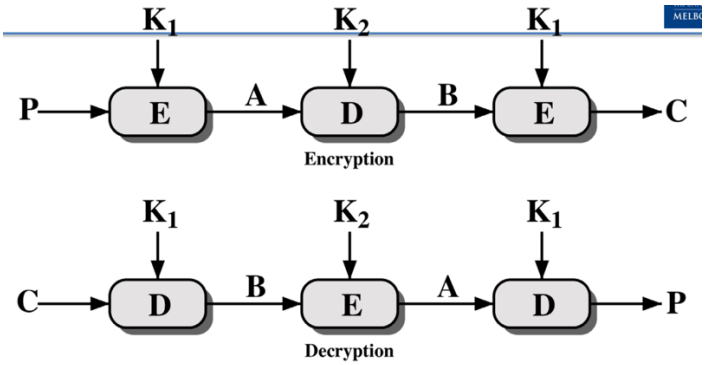


From:William Stalli

  (v) DES: block size = 64, key size = 56, number of rounds = 16.

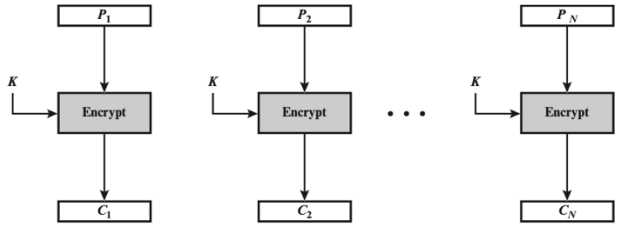
(vi) Strengthening DES: $C = K\_O \oplus DES(K, M \oplus K\_I)$

(vii) Triple DES, AES(128-bits key size).

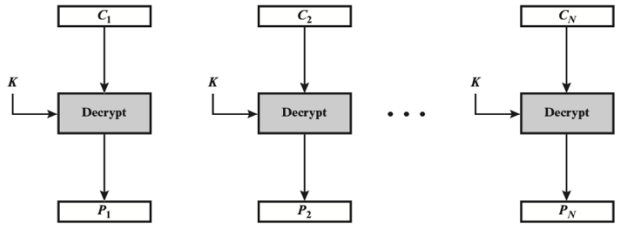


Modes of Block Cipher:

- Electronic Codebook (ECB):

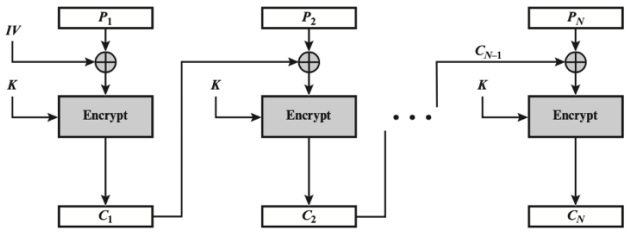


(a) Encryption

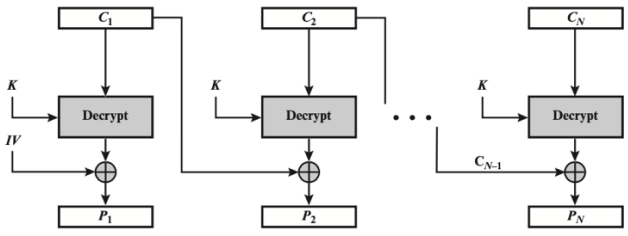


(b) Decryption

- Cipher Block Chaining (CBC):

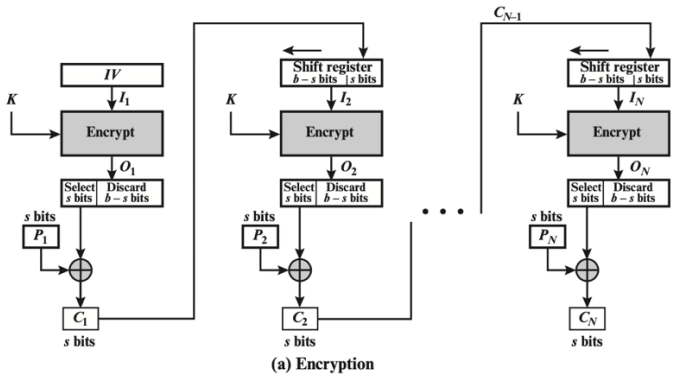


(a) Encryption

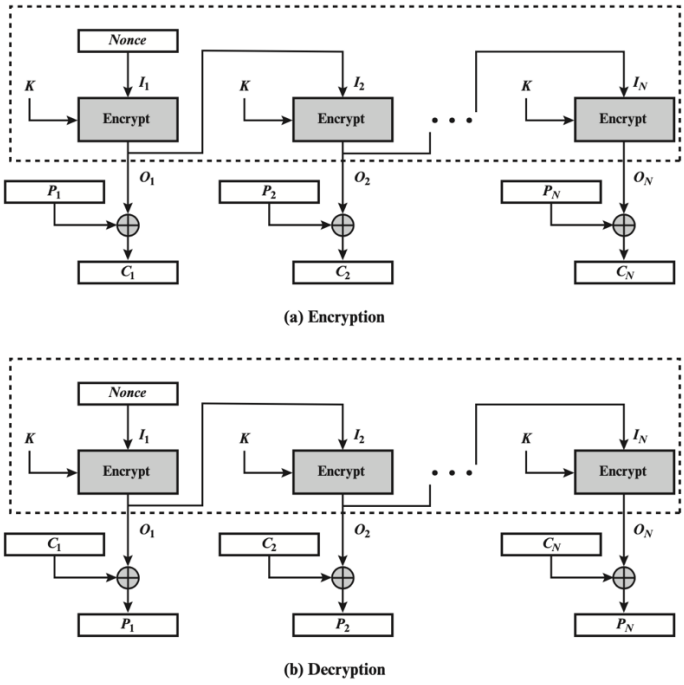


(b) Decryption

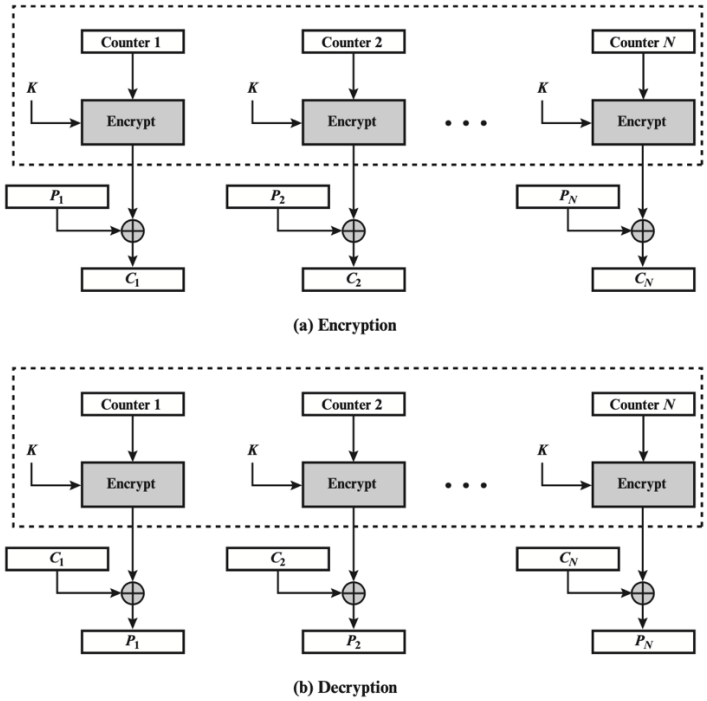
- Cipher Feedback (CFB):



- Output Feedback (OFB):



- Counter (CTR):



Disadvantages of Symmetric Key:

- One key is used for both encryption and decryption.
- Further the key need be shared by both sender and receiver.
- If the key is disclosed, the scheme is compromised.
- Non-repudiation is impossible as sender and receiver are equal. One party can forge other party's data. Hence it does not protect the sender from a receiver forging a message and then claiming that it is sent by the sender.
- In networked situation, the requirement for the key storage grows quadratic in n, the numbers of users. The number of common keys is n(n-1)/2.

Euler's Theorem:

Theorem

If $a \in \mathbf{Z}_n^\star$, *then* $a^{\phi(n)} = 1 \pmod{n}$.

When n equals the multiplication of two prime numbers:

Theorem

If $a \in \mathbf{Z}_{pq}^\star$, *then* $a^{(p-1)(q-1)} = 1 \pmod{pq}$.

Fermat's Theorem:

Theorem

Let p be a prime number, then if $gcd(a, p) = 1$, then

$$a^{p-1} = 1 \ (mod \ p).$$

Fermat's Little Theorem:

Theorem

Let p be a prime number,

$$a^p = a \ (mod \ p), \ \textit{for any integer } a.$$

Group: A group is a set G with a binary operation $\cdot$ on G, such that the following properties hold:

- $\cdot$ is *associative*; that is, for any $a, b, c \in G$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$
- There is an *identity* element e in G such that for all $a \in G$,
$$a \cdot e = e \cdot a = a$$
- For each $a \in G$, there exists an *inverse* element $a^{(-1)} \in G$ such that
$$a \cdot a^{-1} = a^{-1} \cdot a = e$$
- If the group also satisfies
For all $a, b \in G$,
$$a \cdot b = b \cdot a$$
then the group is called *abelian* (or *commutative*).

Ring: A ring is a set R with two binary operations $\cdot$ and $+$, such that:

- R is an abelian group with respect to $+$.
- $\cdot$ is associative; that is, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in R$.
- The *distributive laws* hold; that is , for all $a, b, c \in R$ we have $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$

Prime Field:

We note that the set $\mathbf{Z}_p = \{0, 1, \cdots, p - 1\}$, where p is a prime number, satisfies axioms of a field.

- The set is closed under addition.
- Since p is prime number, any nonzero element in $\mathbf{Z}_p$ has an inverse (Use Extended Euclidean algorithm).
- you can verify that additions and multiplications are distributive.

Characteristics of a Field:

Definition

Let F be a field with the multiplicative identity 1 and the additive identity 0. The characteristic of F, sometimes written as $char(F)$, is the smallest integer $n \geq 0$ such that addition of the 1 with itself n times results in 0. i.e $n(1) = 0$.

Characteristics of real and complex field is 0; Prime field is p, where p is a prime number.

Chinese Remainer Theorem:

If $n_1, n_2, \ldots, n_k$ are pair-wise relatively prime integers, k being a positive integer, the system of simultaneous congruences

$$x \equiv a_1 \ (mod \ n_1),$$
$$x \equiv a_2 \ (mod \ n_2),$$
$$x \equiv a_3 \ (mod \ n_3),$$
$$\ldots$$
$$x \equiv a_k \ (mod \ n_k),$$

has a unique solution modulo $n = n_1 \ n_2 \ \ldots \ n_k$.

Let

$$N_i = n/n_i$$

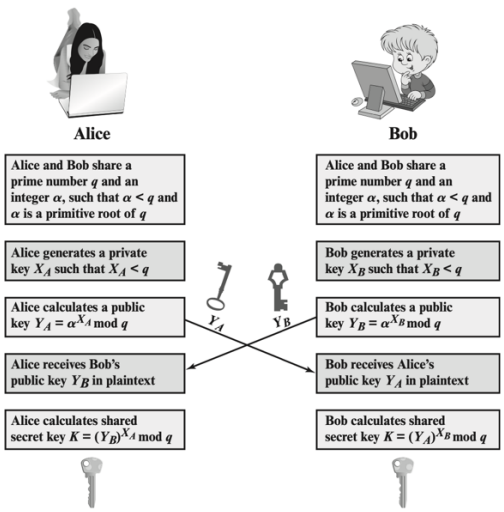
for $i = 1, 2, \ldots, k$.
Choose

$$M_i \ = \ (N_i)^{-1} \ (mod \ n_i),$$

for $i = 1, 2, \ldots, k$.
Then the solution is given by

$$x = \sum_{i=1}^{k} a_i \ N_i \ M_i \ \ (mod \ n).$$

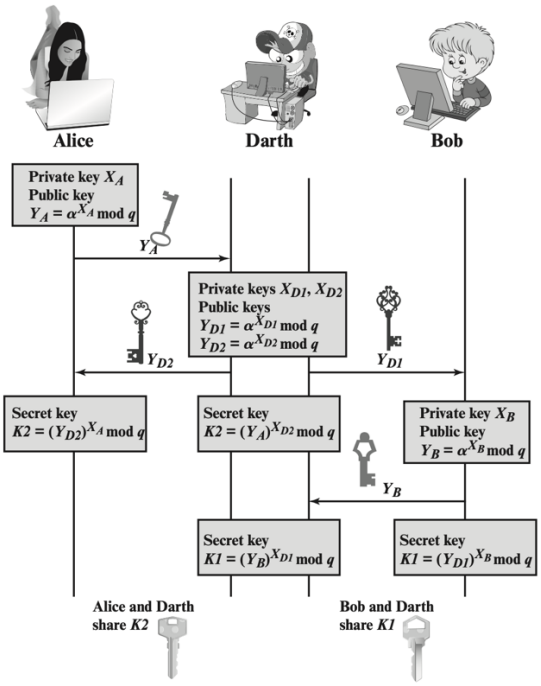
Diffie-Hellman Key Exchange:



One Way Function: Given x, it is efficient to compute f(x), but given f(x), it is infeasible to reverse x.

Discrete Logarithm Problem: Given a, b and n, find x such that: $a^x \bmod n = b$.

Man-in-the-Middle Attack:



RSA Algorithm:

- Let $n = p \times q$; p, q are primes. Let the plain text and cipher text belong to integers modulo n and let (e, d) pair be computed such that

$$e \times d \equiv 1 \bmod \phi(n)$$

(ϕ :Euler's totient function)

- For the RSA key parameter set $K = (n, p, q, e, d)$, define

$$E_k(x) = x^e \bmod n$$

And

$$D_k(y) = y^d \bmod n,$$

where $(x,y \text{ in } Z_n)$. The values (n,e) are termed the public key, and the values p, q and d form the private key.

RSA Problem and Integer Factorization Problem:

- 1. **Integer Factorization problem**: Given a large positive integer n, find its prime factorization. (Every number n can be expressed as $p_1^{e1}p_2^{e2} \ldots p_k^{ek}$, where the pi's are distinct primes and each ei > 1). In particular, if a number n is constructed as a product of two large primes, it is difficult to factor n.
- 2. **RSA problem**: Given a positive integer n that is a product of two distinct odd primes p and q,(n=pq) and a positive integer e such that gcd(e,(p-1)(q-1)) = 1,and an integer c, find an integer m such that

  $m^e = c \pmod n$.

RSA Signature:

N = P×Q; P, Q Large Primes,

Choose Public key e and private key d such that e *d ≡ 1 mod ϕ(N)

Public address – [N, e]
Private address –[d]

Signature Generation:
Message 0 < M < N;
Compute: $s = M^d \bmod N$;
Signature—[M,s]
Verification – if $s^e \bmod N$ =?= M then "Signature Valid"
Else "Signature Invalid"

Issues:
Multiplicative property of RSA signature

$$(M_1 \times M_2)^d = M_1^d \times M_2^d = (M_1 \times M_2)^d$$

i.e. if s_1 = Signature of M_1;
s_2 = Signature of M_2;

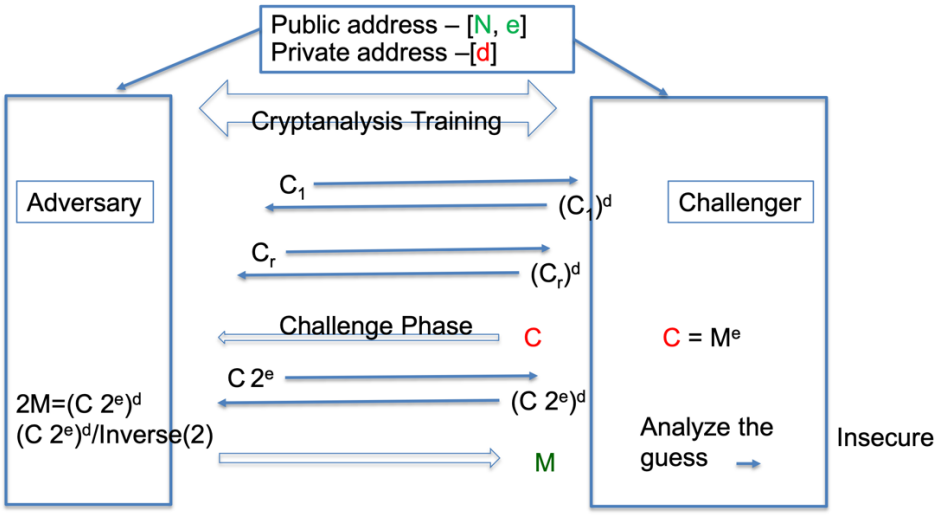
Then $(s_1 \times s_2)$ is the signature of $(M_1 \times M_2)$. Follows from exponential law.

This property leads to a possibility of forgery of signature!
This is in fact an example of existential forgery.

Blinding:

Choose a random x – in the range [0..N-1]
Form a blinded message -- $M_b = x^e M \bmod N$
Alice may agree to sign this blinded message M_b (assume),
Alice then signs the message M_b as $s_b = M_b^d \bmod N$
This blinded signature can be used to compute the signature for M using the multiplicative property:
Now you can compute signature for M as
$s = s_b/x \bmod N$
This is true because, let us apply the verification rule, Note
$s^e = s_b^e / x^e = (M_b)^{de} / x^e = (M_b) / x^e = x^e M / x^e = M$
Hence s is the signature of M. So we have produced a forgery!

Chosen Ciphertext Attack Model in RSA:



Timing Attack:

Here the attacker will observe the behaviour of the Cryptographic algorithms to different inputs and use the experience to break the secret directly.

It can be devastating especially because adversary only needs ciphertexts.

The attack is applicable wide range of cryptographic algorithms.

If you observe variability in any aspects of the crypto algorithm, you may be able to convert into an attack. The generalizations of this attack include power analysis attack and fault based attack. The later, a certain faults are introduced deliberately and attacker studies the algorithm.

Counter measures:

Constant time: One way is to make sure that your algorithm takes a constant time for all inputs. This approach requires you to estimate the longest delay in advance and use appropriate idle time when results take less than the worst case time. However, this method may still leak power profile. In general performance decreases in efficiency.

Random delay: You will add a random delay to algorithm execution to ensures that the relationship between key and the execution time is uncorrelated.

Blinding: You can use the blinding technique introduced earlier. With this, the algorithm takes a random amount time and assures that the relationship between key and the execution time is uncorrelated.

Unkeyed Hash Function (Modification Detection Code):

- What is hash function?

In general, the function takes a variable-length data block as input and produces a fixed length tag or digest satisfying certain properties.

The main objective is to obtain data integrity.

It is referred as unkeyed primitive as does not require any key.

As assumed in the other cryptographic functions, the definition of Hash function is also public.

Hash is also referred to as message digest.

- Integrity:

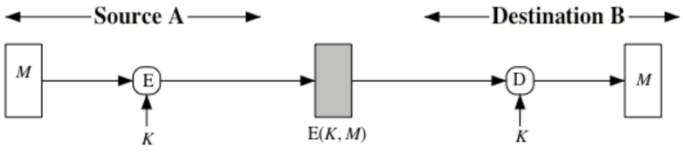
The function Hash has a property that a small change in the message introduces unpredictable changes in the hash value, h = Hash (M).

If a message is changed while in transit, then running Hash function at the received message tells you how the value is deviated from the hash value computed at the source, thus assuring integrity with high probability.

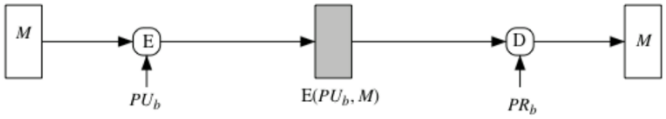
Any intermediate adversary could change the data and compute the hash again. Thus the modification done by Malice cannot be detected.

- Authentication:

To provide authentication with Hash function, you need to involve other encryption techniques, for example, Alice and Bob need to protect the hash value using encryption techniques.

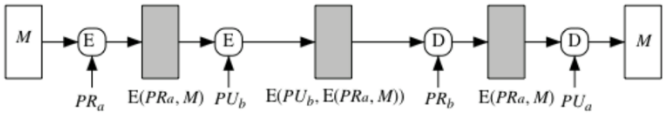


(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality

(c) Public-key encryption: authentication and signature



(d) Public-key encryption: confidentiality, authentication, and signature

- Properties:

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	$H(x)$ is relatively easy to compute for any given x, making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h, it is computationally infeasible to find y such that $H(y) = h$.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) with $x \neq y$, such that $H(x) = H(y)$.
Pseudorandomness	Output of H meets standard tests for pseudorandomness.

Birthday Attack: Consider random numbers from 1 to N using uniform distribution, then the probability that any two numbers are the same exceeds 0.5 after roughly about square root of N trials.

Keyed Hash Function (Message Authentication Code):

- How to achieve?
  At a basic level, we can create a message authentication code using a secret key.
  At a higher level, the keys are carefully managed to obtain higher level guarantees on the exchanged message including source authentication.

- Symmetric key authentication:
  How authentication is obtained?
  - Since they share the key, receiver is sure that the message was created by the sender.
  - By relying on format and structure of the messages, they can detect any modification,
- Public key authentication:

  Public key by nature, anyone can use.

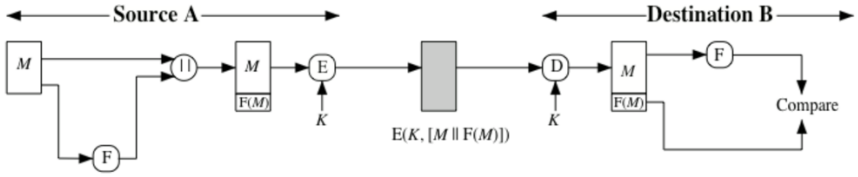
  Does not provide any guarantee for the sender.

  To provide authentication, a sender needs to sign as well (use private key) which can be verified by others using the public key.

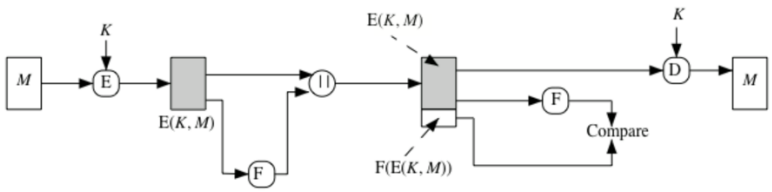
  How do we decide if the message stream is corrupted or not?

  You need some general formatting rules.

- Internal error control vs external error control:



(a) Internal error control



(b) External error control

- Properties:

  MAC has many properties similar to Hash.

  mac:= MAC(Key, message).

  You can treat it as a cryptographic checksum/digest: It takes a arbitrary length message as input and outputs a fixed length authenticator using a key.

  Like hash functions, it is many-to-one function with Preimage resistance (PR).

  For every key, it satisfies hash function properties.

  So sometimes, MAC is referred to as a family of Hash functions.

- Attacks:

  (i) Brute Force

  (ii) Attacker may first determine the key, then he can produce MAC value for any message.

  Sometimes, he may just try to determine a valid tag for a given message.

- Pseudorandom generation: Given a seed number, the pseudorandom number outputs a sequence of random digits.

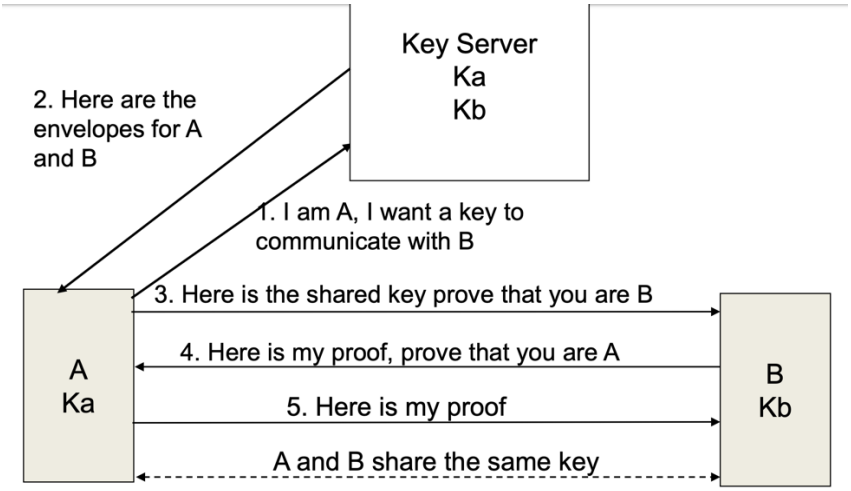
Keys in General:

Keys are to be formed using purely random sources, but in practice, they are usually pseudo random based on some secret seeds or random seeds obtained from physical means.

Long life keys should be generated from a truly random source, examples: Thermal noise, time between key strokes etc.

Symmetric Key Distribution:

- Four methods:
  1. A can select key and physically deliver to B
  2. A third party can select & deliver key to A & B
  3. If A & B have communicated previously can use previous key to encrypt a new key
  4. If A & B have secure communications with a third party C, C can relay key between A & B
- Key server:



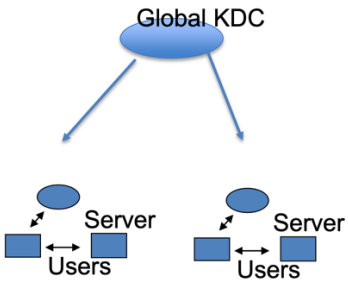
- Session key and master key: session key is a temporary key that used for encryption between users, and it has only one logical session and then discarded. While master key is a long-lasting key that used for encrypting session keys shared by uses and KDC.
- Hierarchical key control:

Key distribution method can be extended to multiple KDCs, a local KDC and a global KDC.

You can have a hierarchy of KDCs.

Users within a same local domain are supported by the local server,

Users in two different domain will need involve global KDC to exchange keys.



Hierarchy of keys minimizes complexity of key distribution. Also localizes the risk of fault or compromise within a local domain.

- Decentralized key control:

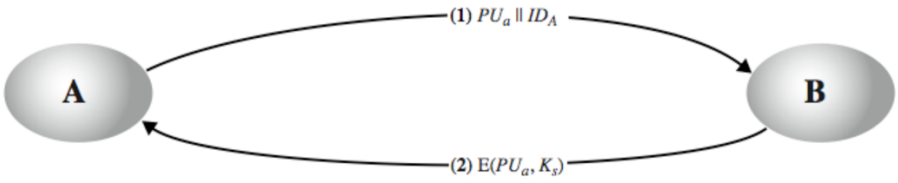
  In the previous method we need to trust KDC which needs a protection from being compromised.

  A fully decentralized approach may require every node establish a master key with every other node, thus needs n(n-1)/2 keys for n end point system.

  A session key algorithm is explained in Fig 14.5 of the textbook.

- Merkle key distribution:

  1. A generates a public/private key pair [*PUa, Pra]* and transmits a message to B consisting of PUa and an identifier of [A, IDA]
  2. B generates a secret key, Ks, and transmits it to A, encrypted with A's public key.
  3. A computes D(PRa, E(PUa, Ks)) to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of Ks.
  4. A discards PUa and PRa and B discards PUa.



Easy to compromise by man-in-the-middle attack.

Random Number:

- Properties:
  - Having Uniform distribution, Statistically random, independent
  - Unpredictability of future values from previous values
- Linear congruential generator:

  $X_{n+1} = (aX_n + c) \bmod m$,
  Where X_0 is the seed and a and c are constants.
- Blum blum shub generator:

  $x_i = LSB(x_{i-1}^2) \bmod n$, where n=p.q, and primes p,q=3 mod 4, LSB: Least Significant Bit.
- Natural random noise:

  There are many events which look random: adiation counters, radio noise, audio noise, thermal noise in diodes, leaky capacitors, mercury discharge tubes etc.

Public Key Distribution:

- Public announcement:

  A simple strategy, users distribute to those who need by any means (broadcasting or email etc)

  Example: PGP keys

  Main issue is that they can be easily forged as we explained before.

- Publicly available directory:

  A directory service is established,
  Each user contacts the directory through secure means and places his public address to be downloaded by other users.
  Each user can update his public key and details. Think, why do you nee this feature?
  Sometime keys may be compromised.
  Users can contact the directory electronically.
  Security is better than the previous method, but still vulnerable..

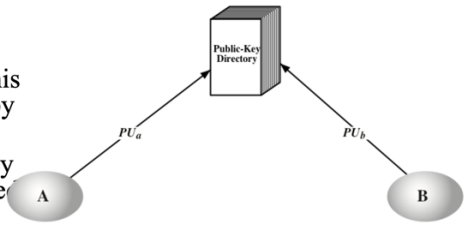


Figure 14.11 Public Key Publication

- Public-key authority:

  This method is a further improvement to the directory service. It has following properties:

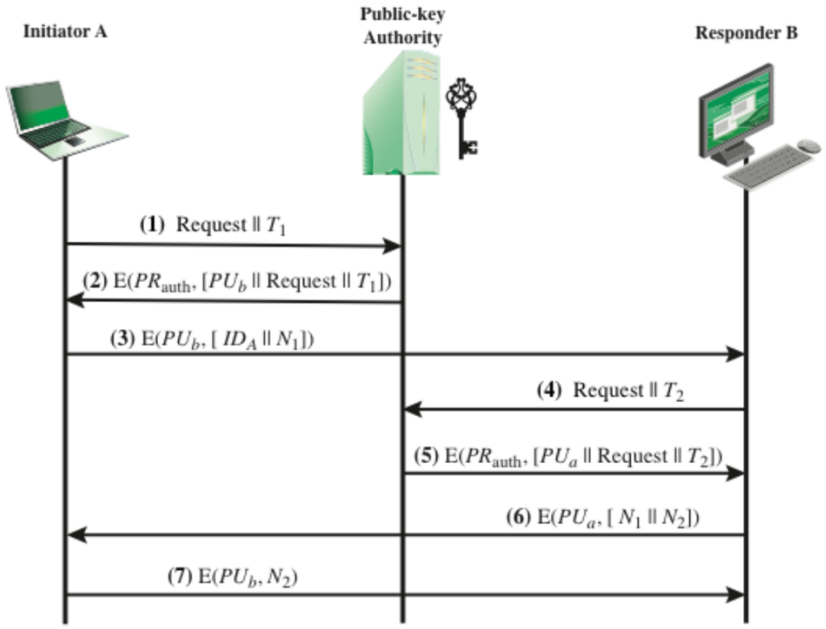
  The authority server is always online with tight control over the distribution and maintenance of keys.

  Authority also has a public and private key: <PU_{auth} ,PR_{auth} >

  Users will contact the authority whenever they need key service.

  Issues:

  – Server needs to be online always.
  – Still there is a possibility of tampering and attacks.



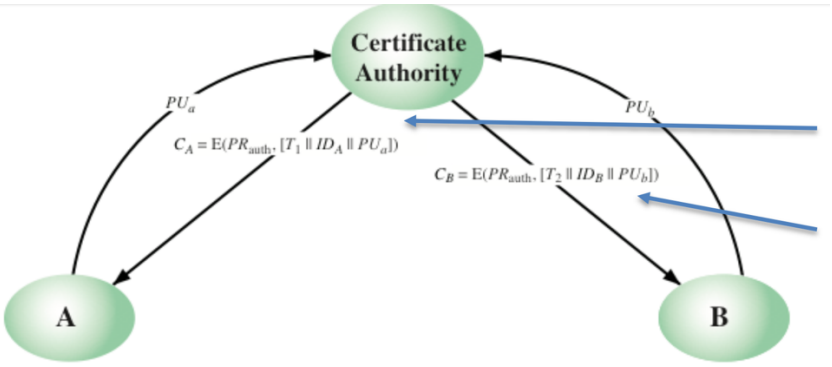
- Public-key certificates:

  Here, key authority need not be online all the time, at least theoretically.

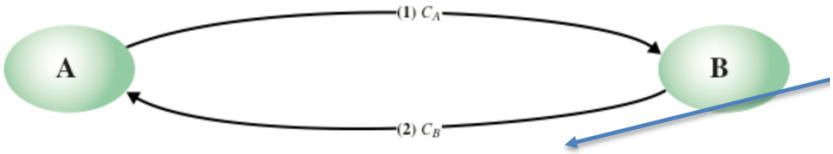
  What is a certificate?

  A form which binds identity of a users with its public key.

  The method allows others to verify the validity of the certificates.

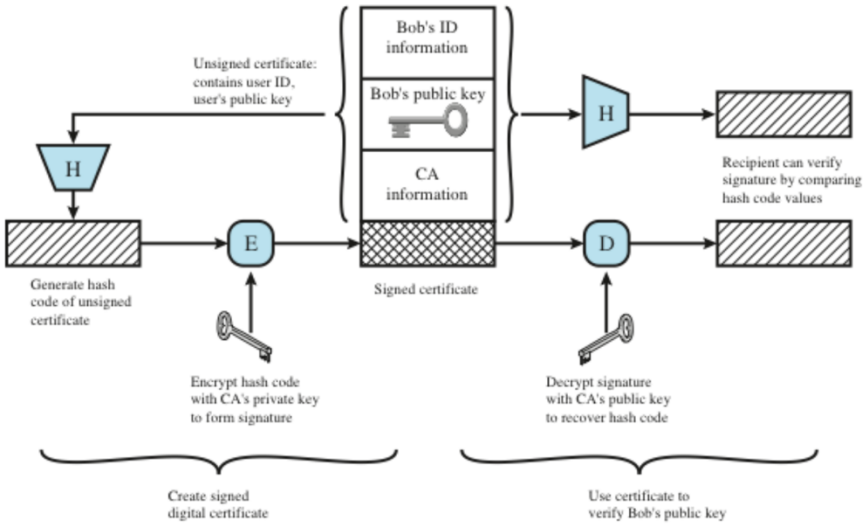


(a) Obtaining certificates from CA



(b) Exchanging certificates

X.509 Certificates:

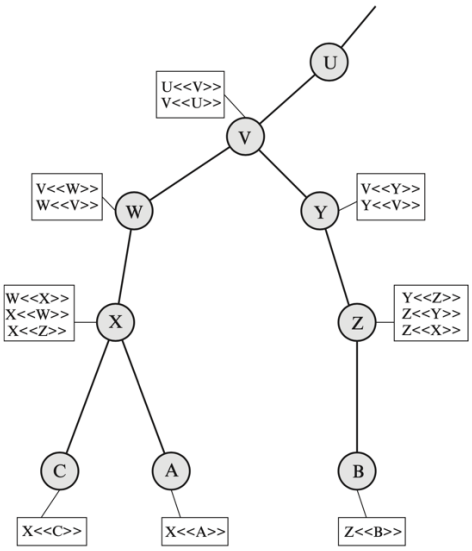


Certificate Authority Hierarchy:

Forward certificates: Certificates of X generated by other CAs, and

Reverse certificates

Certificates generated by X that are the certificates of other CAs.



A acquires B certificate using chain:
X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>

B acquires A certificate using chain:
Z<<Y>>Y<<V>>V<<W>>W<<X>>X<<A>>

ElGamal Encryption:

- Key features:

  DH protocol can be formulated over any cyclic group where computing discrete logarithm over the group is hard.

- Cyclic group:

  Maximum size of cyclic groups obtained from Z_p where p is a prime number: p – 1.

  Maximum size of cyclic groups obtained from Z_n where n is any integer: $\varphi(n)$.

- Encryption:

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

ElGamal Signature:

- Direct digital signature:

  This is attributed to a signature scheme involving only the sender and receiver.

  Here the authenticity of the public key of the source is assured for the destination.

  The scheme is valid depending on the security of the private key. Hence there is a threat that sender could claim that the key is compromised. Such risks could be avoided by having a tighter control on the keys. For example, a requirement of reporting key compromise to a central authority could be included in the policy.

- Essential ideas:

Before proceeding, we need a result from number theory. Recall from Chapter 2 that for a prime number q, if α is a primitive root of q, then

$$\alpha, \alpha^2, \ \ldots \ , \alpha^{q-1}$$

are distinct (mod q). It can be shown that, if α is a primitive root of q, then

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.
2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if and only if $i \equiv j \pmod{q-1}$.

- Signing stage:

1. Generate a random integer X_A, such that $1 < X_A < q-1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

To sign a message M, user A first computes the hash $m = \mathrm{H}(M)$, such that m is an integer in the range $0 \le m \le q-1$. A then forms a digital signature as follows.

1. Choose a random integer K such that $1 \le K \le q-1$ and $\gcd(K, q-1) = 1$. That is, K is relatively prime to $q-1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q-1)$. That is, compute the inverse of K modulo $q-1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1)$.
5. The signature consists of the pair (S_1, S_2).

- Verification stage:

Any user B can verify the signature as follows.

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

- Correctness:

$\alpha^m \bmod q = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$	assume $V_1 = V_2$
$\alpha^m \bmod q = \alpha^{X_A S_1}\alpha^{K S_2} \bmod q$	substituting for Y_A and S_1
$\alpha^{m - X_A S_1} \bmod q = \alpha^{K S_2} \bmod q$	rearranging terms
$m - X_A S_1 \equiv K S_2 \bmod (q-1)$	property of primitive roots
$m - X_A S_1 \equiv K K^{-1}(m - X_A S_1) \bmod (q-1)$	substituting for S_2

Polynomial Rings:

- Polynomials and its properties:

A polynomial over a field F is an expression

$$f(x) = f_n x^n + f_{n-1} x^{n-1} + \cdots + f_1 x + f_0 = \sum_{i=0}^{n} f_i x^i,$$

where the symbol x is an indeterminate and the coefficients $f_i, 0 \leq i \neq n$ are elements of the field. Facts:

- the zero polynomial is $f(x) = 0$.
- The degree of a polynomial $f(x)$, denoted $degf(x)$, is the largest index of a nonzero coefficient. For example, $deg(1 + x + 2x^3)$ is 3, and $deg(1) = deg(1\ x^0) = 0$.
- the degree of a nonzero polynomial is always finite.
- By convention, the degree of the zero polynomial is $(-\infty)$.
- A polynomial of degree n is monic if its leading coefficient f_n (the coefficient of the largest index) is equal to 1. For example, $(1 + x + 2x^3)$ is not monic, however the polynomial $(1 + x + x^3)$ is monic.
- Two polynomials $f(x)$ and $g(x)$ are equal if the coefficients $f_i = g_i$ for all i.
- Set of all polynomials over a field F is denoted $F[x]$.

- Some facts:
  - If a polynomial $r(x)$ divides another polynomial $s(x)$, we say $r(x)|s(x)$, or $s(x)$ is divisible by $r(x)$ or $r(x)$ is a factor of $s(x)$, when $r(x)a(x) = s(x)$.
  - A nonzero polynomial $p(x)$ that is divisible by $p(x)$ or by α, where α is an arbitrary field element, is called an irreducible polynomial.
  - A monic irreducible polynomial is called a prime polynomial.
  - $GCD[r(x), s(x)]$: Greatest common divisor of two polynomials $r(x)$ and $s(x)$, is the monic polynomial of the greatest degree that divides both of them.
  - If the $GCD[r(x), s(x)]$ is 1 then the polynomials $r(x)$ and $s(x)$ are relatively prime.
  - $LCM[r(x), s(x)]$: Least common multiple of two polynomials $r(x)$ and $s(x)$, is the monic polynomial of the smallest degree that is divisible by both of them.

- Galois field:
  It is formally represented as the set of all residues of polynomials in $\mathbf{GF}(p)[x]$ obtained when divided by a prime polynomial $m(x)$ of order k:
  $$\mathbf{GF}(p^k) = \mathbf{GF}(p)[x] \ mod \ m(x),$$
  where $m(x)$ is an irreducible polynomial of degree k. Sometimes, we denore $\mathbf{GF}^{\star}(p^k)$ to denote all non-zero elements of $\mathbf{GF}(p^k)$.
- Example:

i	Elements:x^i	As Polynomials	As Vectors
$-\infty$	0	0	$[0,0,0]$
0	1	1	$[1,0,0]$
1	x	x	$[0,1,0]$
2	x^2	x^2	$[0,0,1]$
3	x^3	$1+x$	$[1,1,0]$
4	x^4	$x+x^2$	$[0,1,1]$
5	x^5	$1+x+x^2$	$[1,1,1]$
6	x^6	$1+x^2$	$[1,0,1]$
7	x^7	1	$[1,0,0]$

Table: Elements of $\mathbf{GF}(2^3)$ as powers of x

i	Elements:x^i	As Polynomials	As Vectors
$-\infty$	0	0	$[0,0]$
0	1	1	$[1,0]$
1	x	x	$[0,1]$
2	x^2	$1+x$	$[1,1]$
3	x^3	$1+2x$	$[1,2]$
4	x^4	2	$[2,0]$
5	x^5	$2x$	$[0,2]$
6	x^6	$2+2x$	$[2,2]$
7	x^7	$2+x$	$[2,1]$
8	x^8	1	$[1,0]$

Table: Elements of $\mathbf{GF}(3^2)$ as powers of x

User Authentication:

- Identification step:
  Identification step: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
- Verification step:
  Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

- Means of authentication:

  Something the individual knows: Examples:a password, a personal identification number (PIN), or answers to a prearranged set of questions.

  Something the individual possesses: Examples: cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token .

  Something the individual is (static biometrics): Examples: Recognition by fingerprint, retina, and face.

  Something the individual does (dynamic biometrics): Examples: recognition by voice pattern, handwriting characteristics, and typing rhythm.

- Replay attack:

  1. The simplest replay attack is one in which the opponent simply copies a message and replays it later
  2. An opponent can replay a timestamped message within the valid time window
  3. An opponent can replay a timestamped message within the valid time window, but in addition, the opponent suppresses the original message; thus, the repetition cannot be detected
  4. Another attack involves a backward replay without modification and is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content

- Approaches of dealing with replay attack:

  Attach a sequence number to each message used in an authentication exchange
  - A new message is accepted only if its sequence number is in the proper order
  - Difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant it has dealt with
  - Generally not used for authentication and key exchange because of overhead

  Timestamps
  - Requires that clocks among the various participants be synchronized
  - Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time

  Challenge/response
  - Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value

- One-way authentication: Email box authentication.

- Mutual identification protocol:

  (i) Goal:

  Goals of such protocols is to provide proof for the communicating parties each other's identity and to exchange session keys for subsequent interactions.

  (ii) Issues to solve:

  1. **Confidentiality**: the exchanged session keys are protected,
  2. **Timeliness**: Ensure that the exchange is current and prevent replay attacks.

(iii) Needham-Schroeder protocol:

1. $A \rightarrow KDC$: $ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A$: $E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B$: $E(K_b, [K_s \| ID_A])$
4. $B \rightarrow A$: $E(K_s, N_2)$
5. $A \rightarrow B$: $E(K_s, f(N_2))$ where f() is a generic function that modifies the value of the nonce.

(iv) Denning modification:

1. $A \rightarrow KDC$: $ID_A \| ID_B$
2. $KDC \rightarrow A$: $E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B$: $E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A$: $E(K_s, N_1)$
5. $A \rightarrow B$: $E(K_s, f(N_1))$

(v) Neuman modification:

1. $A \rightarrow B$: $ID_A \| N_a$
2. $B \rightarrow KDC$: $ID_B \| N_b \| E(K_b, [ID_A \| N_a \| T_b])$
3. $KDC \rightarrow A$: $E(K_a, [ID_B \| N_a \| K_s \| T_b]) \| E(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $A \rightarrow B$: $E(K_b, [ID_A \| K_s \| T_b]) \| E(K_s, N_b)$

Hybrid Encryption:

Use public key, such as RSA to share a secrete key and encrypt a large file by using secrete key (AES).

SSL:

Secure Socket layer protocol uses Transport Later features of Modern Internet.

The main idea is to create a transport session between two nodes and then exchange a session key using a protocol similar to the Hybrid protocol.

Session key is used in the symmetric key encryption.

So, a Transport Layer Security (TLS) used two important concepts:

- Connection between a client and a server
- Session associated with the connection.

They use OSI layering model protocols for realizing the above concepts.