Information & Network Security (Attacks, Data Encryptions, and Defense Mechanisms)

Prof. K.R. Chowdhary, Director JIET-SETG
Email: kr.chowdhary@jietjodhpur.ac.in
Webpage: www.krchowdhary.com

JIET School of Engineering & Technology for Girls (SETG)

Saturday 3rd October, 2015

Information Network

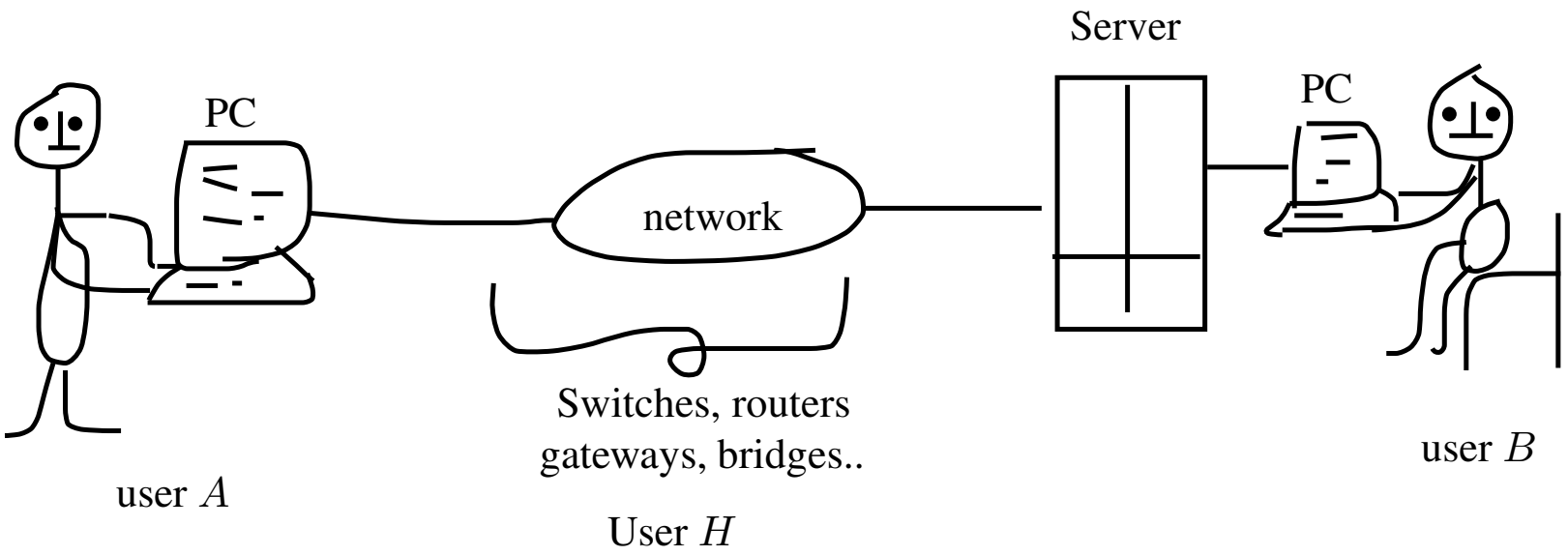


Figure 1: Information Network

Security Attacks

Information and network security risks are increasing with the growth of the number of threats and sophisticated attacks.

- A user U_A, transmits a file to U_B. User U_H (hacker) captures a copy of the file during transmission.
- A network manager Man_D, transmits a message to a networked computer Com_E to grant accesses / authorizations for new users. U_H intercepts the message, alters it and forwards to Com_E.
- A message sent from customer U_A to stock-broker U_B to execute some transactions. U_H intercepts and sends several copies.
- An employee Y is terminated. The personal manager sends message to server to invalidate $Y's$ account. Y delays the message, and retrieves the sensitive information before message reaches.

Aspects of security needs

Need of a systematic way of defining the requirements for security and characterizing the approaches to satisfy the requirements.

- Identify the Security attacks
- Identify the Security mechanisms
- Identify the Security services

Security Services

- Authentication
- Confidentiality
- Integrity
- Non-repudiation
- Access-control
- Denial of Service (availability Function)
- Spoofing

Common Attacks

- Eavesdropping
- Interception may be caused due to eaves dropping
- Cryptoanalysis
- Password Pilfering
  1. Guessing
  2. Social Engineering
  3. Phising (Mass social engineering attacks, the main form are disguised emails, as if these emails were sent by banks, credit card holders, etc. In reply we supply the information, which go to wrong hands.)
  4. Dictionary attacks
  5. Password Sniffing: These are programs, which capture remote login information such as user names and user passwords.

Defense Mechanisms

- Password Protection.
  1. Use long passwords with combinations
  2. Do not reveal password to any one
  3. Change password periodically
  4. Do not use same password for different accounts
  5. Do not use remote logins
  6. Avoid entering any information in any popup window
- User authentication methods:
  1. Using passwords
  2. biometrics,
  3. Electronic passes authenticated by the issuer.

Common Attacks and Defense Mechanisms

- Identity Spoofing: Allows attackers to impersonate a victim without using the victim's passwords. Following are common attacks:
  1. man-in-the-middle attacks

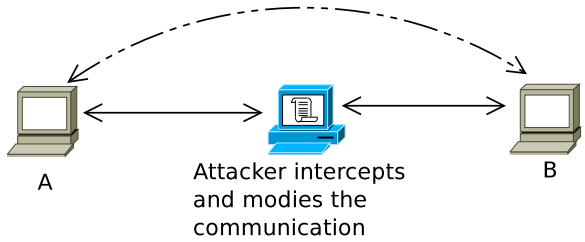


Figure 2: Man-in-the-middle attacks.

  2. Message Replays
- Common mechanisms to thwarting message replay attacks:
  1. Attach a random number to the message
  2. Attach a time stamp to the message
  3. Use prime number and time stamp together

Common Attacks and Defense Mechanisms

- Network Spoofing: IP spoofing is one of the major network spoofing technique. Following are the types:
  1. Syn Flooding: Exploits the implementation side effect of the TCP/IP network protocol.
  2. TCP hijacking
  3. ARP Spoofing
- Identity Spoofing:
  1. Message Replays
  2. Buffer overflow Exploitation

```
int main(){ char buffer[8];
    char *str = "This is a test of buffer overflow.";
    strcpy(buffer, str);
    printf("%d", buffer); }
```

Protections: Security policies

- Security policies
  1. Access policy
  2. Accountability policy
  3. Authentication policy
  4. Availability Policy
  5. Maintenance policy
  6. Violation reporting policy

Protection of users and networks

- Protection of Employees
  1. Shared key encryption schemes
  2. Hash Functions. Hash function H generates $H(m)$ for every variable-length message m. $H(m)$ is message digest
  3. Digital Signature: Binding an information to an originator.
- Protection of Networks
  1. Firewalls
  2. Access, authorization, and authentication tools
  3. Intrusion detection systems

Cryptography

- Method of sending messages in disguised form, so that only the intended recipients can remove the disguise and read the messages.
- Terms:*plain text, cipher text, cryptosystem*

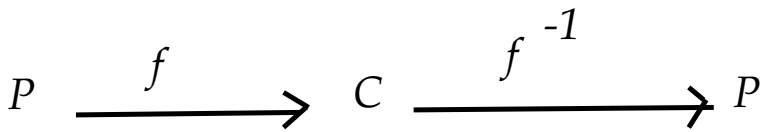


Figure 3: Cryptosystem

- To facilitate rapid enciphering/deciphering, there should be simple rules.
- For example: $A-Z \Longleftrightarrow 0-25$. Let $p \in \{0,\ldots,25\}$ is plain-text, and f is from $\{0,1,\ldots,25\}$ to itself:

Some Examples of secrecy systems

- Simple Substitution Cipher (Letter of the message is replaced by a fixed substitute) $M = m_1 m_2 \ldots$, where m_1, m_2 are the successive letters, becomes: $E = e_1 e_2 \cdots = f(m_1) f(m_2) \ldots$
- Transposition (Fixed Period d): The message is divided into groups of length d and a permutation applied to the first group, the same permutation to the second group, etc.
  Thus for d = 5, we might have 2 3 1 5 4 as the permutation. This means that: $m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8 m_9 m_{10} \ldots$ becomes $m_2 m_3 m_1 m_5 m_4 m_7 m_8 m_6 m_{10} m_9 \ldots$

Data Encryption Algorithm design Criteria

- XOR Function: $X \oplus Y = (x_1 \oplus y_1)(x_2 \oplus y_2)\ldots(x_n \oplus y_n)$
- We use E, D and K to denote encryption algorithm, a decryption algorithm, and secret key, respectively.
- Plaintext M is divided into sequence of blocks: $M_1, M_2, \ldots, M_k$, where each block is $\ell-$bit long, except possibly the last block M_k. If $|M_k| < \ell$, add an 8-bit control code at the end of M_k, once or several times to obtain a new block of size ℓ. This is called *padding*.
- An encryption algorithm which encrypts a block at a time is called block-cipher algorithm.
- When $\ell = 8$ we call it *stream-cipher* algorithm.
- The encryption algorithm encrypts M_i to produce a ciphertext block C_i is:

$$C_i = E_k(M_i) = K \oplus M_i \tag{1}$$

Data Encryption Algorithm design Criteria

- The decryption algorithm decrypts C_i into M_i is as follows:

$$D_k(C_i) = K \oplus C_i = K \oplus (K \oplus M_i) = (K \oplus K) \oplus M_i = 0^\ell \oplus M_i = M_i \quad (2)$$

- The XOR is the simplest encryption algorithm. For example, $\ell = 16$, $K = 100110101001101 1$, then E encrypts 'FUN' as follows:

plaintext	F	U	N	(padding)
ASCII	01000110	01010101	01001110	00001010
secret key:$\oplus$	10011010	10011011	10011010	10011011
ciphertext:	11011100	11001110	11010100	10010001

- The XOR algorithm is simple and fast, but resulting level of security is low. The eavesdroppers can easily calculate the secret key K from a plain text-cipher text pair as follows:

$$M_i \oplus C_i = M_i \oplus (M_i \oplus K) = K. \quad (3)$$

Data Encryption Algorithm design Criteria

- Attacks like this derive secret keys using a small number of samples of cipher-text data and corresponding plain text, are called *known-plain text attacks.*
- Therefore, the key must be frequently changed. If key is used once only, the XOR provides the best security. but, this requires large number of keys.
- Think of other problems of one-time pad?

Criteria for Data Encryption

Qualities of encryption/decryption algorithms

- Must be easy to implement on hardware and software.
- The time and space complexity of both encryption and decryption must be small and constant factor of input.
- Only those operations must be employed, which are easy to implement on computer.
- Following operations are common: OR, AND, substitution, ex-OR, shift-R/L, permutations, unary operations, etc.

Criteria for Data Encryption

Resistance to statistical analysis

- The encryption algorithm must destroy any statistical structure in the plain text data. This is satisfied by the diffusion and confusion properties of the algorithm.
- diffusion: Changing a single bit in plain text will cause a number of bits in cipher text to be changed. These should be distributed.
- confusion: change of single bit in key will cause number of bits in the cipher text to be changed, and evenly distributed.
- Diffusion is achieved by repeatedly executing fixed sequence of operations for a fixed rounds on strings generated from the previous round.
- Confusion may be achieved by generating number of sub-keys from the key K. And, use the subkeys to operate the plaintext in different rounds.

Criteria for Data Encryption

Resistance to Brute-Force Attacks

- If encryption key is ℓ bit long. An eavesdropper could brute force to decipher C by calculating $M' = D_{K'}(C)$ for each ℓ bit string. There are 2^{ℓ} different ℓ-bit strings.
- The ℓ must be large. It is common belief that $\ell = 128$ bit will take many years to break.

Criteria for Data Encryption

Resistance to other attacks:

- Encryption algorithm must resists other attacks. These include chosen plain Plain text attacks and mathematical attacks.
- Chosen Plain text attacks: The attacker choses a plain text message to lure the opponents to encrypt it as a bait. It contains useful information for the attacker.
- Mathematical attacks: Attacker uses the mathematical methods to decipher the cipher text. These are: differential cryptanalysis, linear cryptanalysis, algebraic cryptanalysis.

Symmetric Encryption

- Sender and receiver share the same secret key. Algorithm is called secret key or symmetric key algorithm.
- The encryption assumes that it is computationally unfeasible to decrypt a message given the ciphertext and knowledge of encryption algorithm.
- Secret-key encryption features: Message space M, key space K, cipher space C, and two maps (encryption and decryption).

$$\phi : M \times K \rightarrow C \tag{4}$$

$$\gamma : C \times K \rightarrow M \tag{5}$$

- Both functions must satisfy:

$$\phi(\gamma(c, k), k) = c \text{ and } \gamma(\phi(m, k), k) = m$$

$\phi(-, k) : M \rightarrow C$; One way function for all k $\phi(m, -) : K \rightarrow C$; One way function for all m

One way function

- One way functions: It is easy to compute (compute in polynomial complexity). Infeasible: not computable in polynomial time.
- Secret key must be communicated prior to use, through a secure channel.
- Disadv: Key need to be communicated securely. Hence, requires key distribution.
- Example of secret key: Data Encryption Standard (DES), block based technique, 64-bit long fixed block text, transform it into 64-bit ciphertext. (56-bits are encrypted and decrypted, 8-bits for parity check),
- Total keys: $2^{56} = 7.2 * 10^{16}$.

Model of secret-key cryptosystem

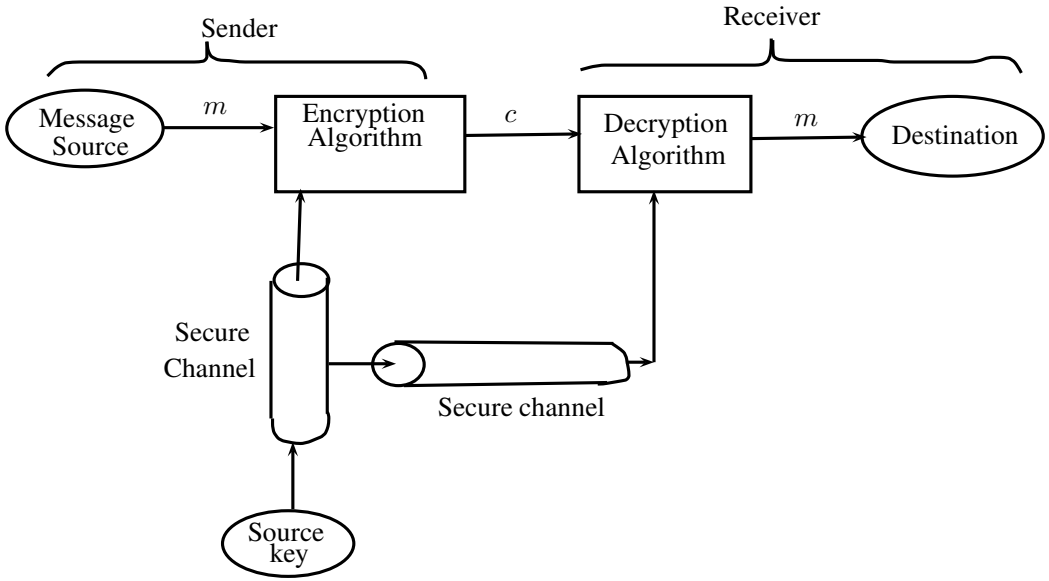


Figure 4: Secret-key Cryptosystem

Table: ciphertext obtained by XOR of m and k.

Plaintext:	s e c u r i t y o f e - s e r v i c e s
Key:	x k l p o r w x r u g e s m y a w f h v
Ciphertext:	k n o e 2 c a - 2 s - b e - n s w 3 e b e

Figure 5: DES encription.

DES Algorithm

- Algo: Series of permutations and substitutions. A block (56-bits) subjected to an initial permutations (IP), followed with complex, key dependent cycles, involving 16-sub keys and functions. Finally, through the permutation IP^{-1}.
- $c_i = m_i \oplus k_i$, for all $i \leq \text{length}(m)$, where $\oplus$ is exclusive OR, i is i^{th} binary bit of plaintext m, k_i is ith bit of key k, c_i is i^{th} binary bit of cipher text c.
- Intermediate halves of ciphertext are comparable to intermediate keys. Also, it provides *non-linearity*.

$$L_i = R_{i-1} \tag{6}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i). \tag{7}$$

DES Algorithm

- Figure shows the main operations executed at each cycle. Each right half expanded from 32 to 48 bits.

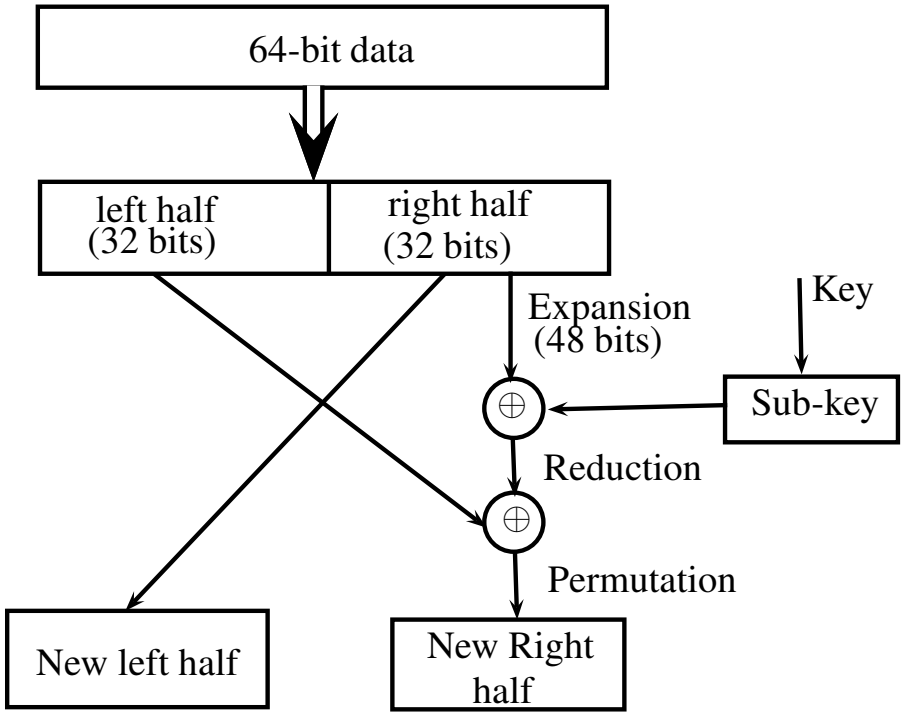


Figure 6: A DES Cycle.

DES Algorithm

- DES Weaknesses: Two messages m and $\overline{m}$ are called complement if $m \oplus \overline{m} = 1111\ldots$. hence if m is encrypted with a k, the complement of the resulting cipher will be the encryption of the complement $\overline{m}$, using the complement key $\overline{k}$
- Semi-key weakness: There are specific pair of keys having identical decryption, implying that two keys decrypt a message encrypted by a single key.
- Can be broken by brute-force attack. Total private key permissions $= 2^{56} = 10^3 10^3 10^3 10^3 2^6 = 64 \times 10^{15}$. If an attacking CPU can generate 64×10^6 permutations per sec., it will take, 10^9 secs. $\approx$ 11,000 days.
- Better algorithms are Triple DES (56*3=168 length), and AES.

Public-key cryptosystem

- User U_b wants to send a secret message to U_a. K_b is public, K_a is secret (in possession of U_a). There is a function:

$$\alpha : C \rightarrow C \text{ such that}$$
$$\gamma(\phi(m, \alpha(k)), k) = m, \text{ and}$$
$$\phi(\gamma(c, k), \alpha(k)) = c$$

- $(\alpha(k), k)$ are pair of private and public keys, and α is link between private and public keys.

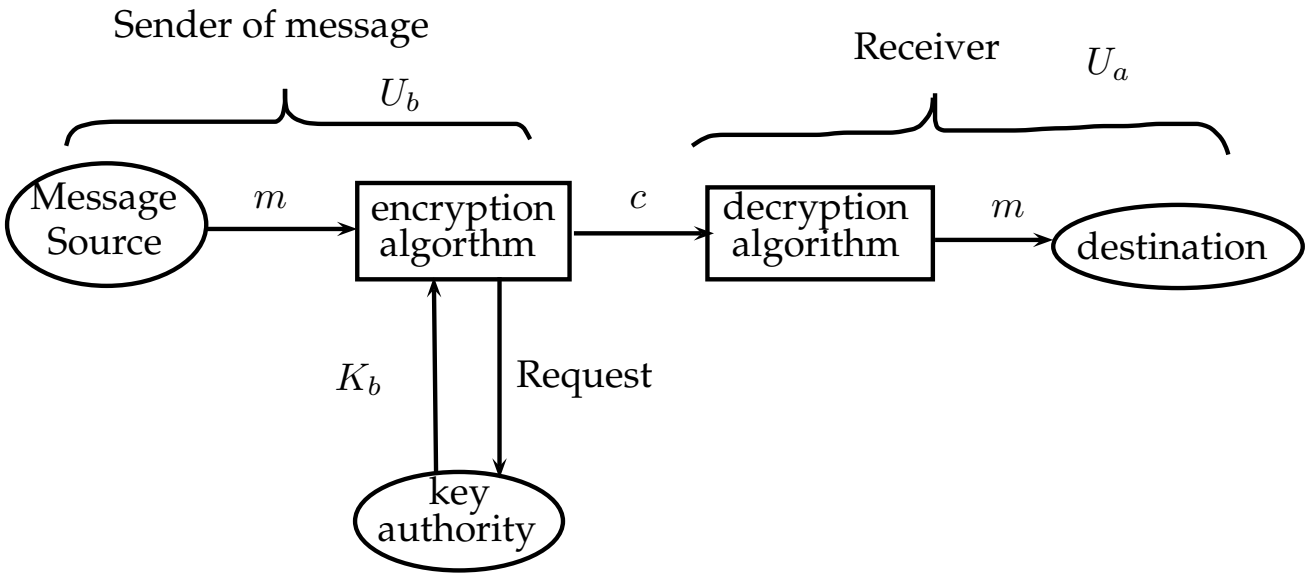


Figure 7: Public-key Cryptosystem

Public-key main algorithm

RSA Algorithm:

- 1978: secure and widely accepted, difficulty to determine prime factor of large integers.
- Block cypher; Plain-text, cipher-text are 0 to $n-1$.
- following requirement must be met:

  It is possible to find integers n, e and d such that:

  $m^{ed} = m \ mod \ n$, for all $0 \leq m < n$. It is relatively easy to compute $m^d \, mod \ n$ and $c^e \, mod \ n$ for $m < n$ but computationally infeasible to compute d given e and n.
- Two interchangeable keys (n, d) (public) and (n, e) (private-key).

Public-key main algorithm

RSA Algorithm: continued..

- Plain-text block m having value less than n is encrypted/decrypted as:
  $c = m^d \text{mod } n$, and $m = c^e \text{ mod } n$.
- e is selected such that: $(m^e)^d \text{ mod } n = m$, $e.d = k\phi(n) + 1$, $\phi(n) = (p-1)(q-1)$ is Euler Function.
  $n = p.q$, where p, q are primes, d is relative prime to $(p-1)(q-1)$, i.e. d has no factors common with $(p-1)(q-1)$
- let $p = 17$, $q = 13$, $\therefore n = pq = 221$, $\phi(n) = 192, d = 5, e = 77$,
- Let $m = 6, \therefore, c = 6^5 \text{ mod } 221 = 41$. is encryption, and $m = 41^{77} \text{ mod } 221 = 6$ is decryption.

Strength of RSA

- RSA is based on discrete log. I.e., "finding of two large primes and multiplying them together to get n is far easier than, going other way, i.e., given n find the primes.
- This fundamental principle of number theory is "trapdoor", i.e., "one-way" entry.
- Given b and x, we can compute b^x by continuous squaring b, easily.
- But, given value of b^x, which is say equal to y, i.e, $y = b^x$, how find x? That is, $x = \log_b y$. This is challenging !
- Say, $y = 2^{16} = 65536$ (easy to compute) !. Now say, given 65536, find $x = \log_2 65536$. This question is "discrete logarithm problem" and is strength of RSA.

Evaluation of Cryptosystem

- Amount of Secrecy
- Size of Key
- Complexity of Enciphering and Deciphering Operations
- Propagation of Errors
- Expansion of Message

Other Formal models (who should receive the info?)

Lattice Model for secure information flow

- Lattice: Finite Set + Partial Ordering relation, such that for every pair there is least upper bound and greatest lower bound.
- $(S, \leq)$ is POSET.

Linear ordered lattice

$$S = \{A_1, A_2, \ldots, A_n\}$$

$$A_i \rightarrow A_j \text{iff } A_i \leq A_j$$

$$A_i \oplus A_j = A_{max}(i, j)$$

$$A_i \otimes A_j = A_{min}(i, j)$$

$$L = A_1; H = A_n.$$

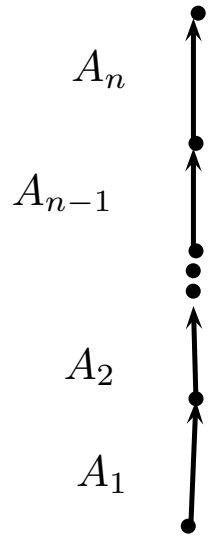


Figure 8: Linear ordered lattice

Nonlinear lattice model (who should receive the info?)

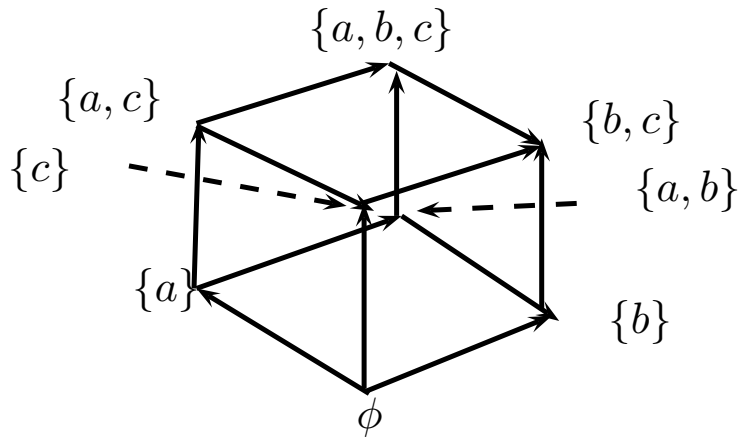
- It is richer structure; classes are hierarchically ordered.
- Suitable for providing security at different levels (Govt. & Military) or objects in programs, and OS.
- Information may be: unclassified, confidential, secret, top secret.

$$A \rightarrow B,\ iff\ A \subseteq B$$

$$A \oplus B = A \cup B$$

$$A \otimes B = A \cap B$$

$$L = \phi; H = S$$



Lattice Structure for $(P(S), \subseteq)$
$S = \{a, b, c\}$

Figure 9: Lattice Structure

Access Matrix Model (who should receive the info?)

- Three principle components: *(1) Set of passive objects*, *(2) set of active subjects* (which manipulate other objects), and *(3) set of rules governing manipulation of objects by subjects*.
- Objects: files, terminals, devices. Subjects: Processes (every subject may be also an object, hence it may be manipulated by other subjects).
- Entry for a row-column reflects the mode of access and ownership for subject-object

Final Words

- No security protection mechanism is perfect yet.
- It is claimed that quantum cryptography, which is based on the quantum state of particles, will be a strong cryptosystem, and shall be impossible to break.
- On the other side, as the processing power of CPUs are increasing constantly, and 100 core processors are not far off, it will be easier to break the encryption easier using multiple desktops using these systems.

Unix tools for security

chmod : sets the permissions for files and directories

chgrp: change group for a files/directory

chown: change owner for a file/directory

groups: lists groups

passwd: change the password for current user

cat /etc/passwd: password file

Unix command *crypt*

- crypt is the password encryption function. It is based on the DES algorithm with variations
- key is a user's typed password.
- *salt* is a two-character string chosen from the set [a-zA-Z0-9./]. This string is used to perturb the algorithm in one of 4096 different ways
- By taking the lowest 7 bits of each of the first eight characters of the key, a 56-bit key is obtained. This 56-bit key is used to encrypt repeatedly a constant string (usually a string consisting of all zeros).
- The returned value points to the encrypted password, a series of 13 printable ASCII characters (the first two characters represent the salt itself). The return value points to static data whose content is overwritten by each call.

Unix command *crypt*

- The key space consists of 2**56 equal 7.2e16 possible values. Exhaustive searches of this key space are possible using massively parallel computers. Software, such as *crack*, is available which will search the portion of this key space that is generally used by humans for passwords.
- Hence, password selection should, at minimum, avoid common words and names. The use of a *passwd* program that checks for crackable passwords during the selection process is recommended.
- RETURN VALUE: A pointer to the encrypted password is returned. On error, NULL is returned.

Unix command *crypt*

- NAME: crypt, mcrypt, mdecrypt - encrypt or decrypt files
- SYNOPSIS: mcrypt [-dLFubhvrzp] [-a algorithm] [-c config_file] [-m mode] [-s keysize] [-o keymode] [-k key1 key2 ...] [-f keyfile] [filename ...]
- mdecrypt [-LFusbhvzp] [-a algorithm] [-c config_file] [-m mode] [-s keysize] [-o keymode] [-k key1 key2 ...] [-f keyfile] [filename ...]
- mcrypt -a des –bare of.c; output: .nc
- mcrypt -d -a des –bare of.c.nc