

MODULE 1

Introduction, Basics of Cryptography, Secret Key Cryptography

- Computer security is all about studying cyber attacks with a view to defending against them.
- The attacks include pharming and phishing attacks together with assorted malware and denial of service attacks.
- Understanding what makes systems vulnerable to these attacks is an important first step in avoiding or preventing them.
- There are different classes of vulnerabilities including those caused by poorly written or configured software.
- There are diverse defence strategies such as Access control, authentication, and data protection techniques are introduced.

1.1 Cyber Attacks

1.1.1 Motives

- "**What are the main goals of an attacker?**"
The sheer thrill of mounting a successful cyber attack has been motivation enough for hackers (Table 1.1).
- Most hackers were (and still are) young adults, often teens, who had dropped out of school but were otherwise intelligent and focused.
- Many of the "**traditional**" hackers seem to be obsessive programmers.
- Often hackers use **scripts and attack kits** designed by others (these can be freely downloaded from the Internet). Their activities do not require any special programming skills or advanced knowledge of computer systems.
- Other perpetrators of cyber attacks include company insiders, often employees who wish to gain illegal access and have extra privileges
- There is also a **serious threat from cyber terrorists**
- **Cyber terrorism** is one weapon which may include biological, chemical, and nuclear weapons. Their goals are to **cripple the information/communication systems of the financial and business institutions of their "enemies."**
- The primary motivation for launching cyber attacks has shifted to **financial gain**.

Table 1 . 1 Notable cyber attacks

Year	Event
1988	Robert Morris, a 23-year-old Cornell graduate student, released a worm that overran

	Arpanet, incapacitating almost 6000 computers, congesting government and university systems. He was fined \$10,000 and sentenced to 3 years probation.
1991	31-year-old David L. Smith created the worm "Melissa," which infected thousands of computers causing damage of approximately \$1.5 billion. This virus sent copies of itself to the first 50 names of the recipient's address book. He received a 20-month jail term.
2001	"Anna Kournikova" virus. Promising photos of the tennis star mailed itself to the every person in the victim's address book. Investigators were apprehensive that the virus was created with a toolkit enabling the rookies to create a virus.
2008	The headquarters of the Obama and McCain presidential campaigns were hacked.

Some of the main motives of launching cyber attacks are:

1. **Theft of sensitive information.**
2. **Disruption of service.**
3. **Illegal access to or use of resources.**

1. Theft of sensitive information.

- Many organizations store and communicate sensitive information.
- Information on new products being designed or revenue sources can be hugely advantageous to a company's competitors.
- Likewise, details of **military installations or precise military plans** can be of immense value to a nation's adversaries.
- Political spying targeted at government ministries and national intelligence can HAVE many sensitive operations planned for the future.
- Besides corporations, banks, the military, intelligence, etc., the individual too has increasingly been a target.
- Leakage of personal information such as **credit card numbers, passwords, and even personal spending habits are common and are collectively referred to as identity theft.** Such information is advertised on certain websites and may be purchased for a small fee.

2. Disruption of service.

- Interruption or disruption of service is launched against an organization's servers so they are made unavailable or inaccessible.
- In recent times, there have been unconfirmed reports of such attacks being launched by business rivals of e-commerce websites.

- The goal here appears to be "*my competitor's loss is my gain.*" In 2001, there were a series of such attacks that targeted the websites of Yahoo, Microsoft, etc. in a short span of time.
- They were meant to alert corporates and others of the dangers of this class of attacks.

3. Illegal access to or use of resources.

- The goal here is to obtain free access or service to paid services.
- Examples of this include free access to online digital products such as magazine or journal articles, free talk time on someone else's account, free use of computing power on a supercomputer, etc.
- In each case, the attacker is able to circumvent controls that permit access to only paid subscribers of such services.

1.1.2 Common Attacks

Some of the common attacks are :

1. **Phishing**
2. **Pharming**
3. **Dictionary attacks**
4. **Denial of Service (dos)**
5. **Trojan**
6. **Spyware**

1. Phishing:

- One set of attacks are those that attempt to *retrieve personal information* from an *individual*.
- It *provokes the victims to a fake website* — an on-line bank, for example.
- The fake site has the look and feel of the authentic bank with which the victim has an account.
- The victim is then asked to enter sensitive information such as his/her login name and password, which are then passed on to the fake website.
- Personal information may also be leaked out from credit cards, smart cards, and ATM cards through a variety of skimming attacks.

2. Pharming:

- It attempts to deduce *sensitive information from lost or stolen smart cards* through advanced power and timing measurements conducted on them.
- Finally, *leakage of information* may also take place through **eavesdropping or snooping** on the link between two communicating parties.

3. Dictionary attacks :

- One means of intruding into a computer system is through password-guessing attacks.
- The ultimate goal of the attacker is *to impersonate his/her victim*.
- The attacker can then perform unauthorized logins (break-ins), make on-line purchases, initiate banking transactions, etc., all under the assumed identity of the victim.

4. Denial of Service (DoS):

- Denial of Service (DoS) means the attacker performs a *interruption or disruption of the computing services on a system* .
- These attacks exhaust the *computing power, memory capacity, or communication bandwidth* of their targets so they are rendered unavailable.
- One version of this attack causes website defacement.
- At various times, the websites of high-profile targets such as the American president or various government ministries have been targeted.
- To prevent such attacks an *alarm* being raised,
- Dos attack on a web server slows down the web server so that its response time to requests from the outside world is unacceptably high.

5. Malware.

- Worms and viruses are malware that replicate themselves.
- A virus typically infects a file, so a virus spreads from one file to another.
- A worm is usually a stand-alone program that infects a computer, so a worm spreads from one computer to another.
- Worms and viruses use various spreading techniques and media — e-mail, Internet messages, web pages, Bluetooth, and MMS are some of the propagation vectors.
- Trojan:A trojan is a kind of malware that masquerades as a utility but has other goals such as the modification of files, data theft, etc.
- Spyware, installed on a machine, can be used to monitor user activity and as a key logger to recover valuable information such as passwords from user keystrokes.

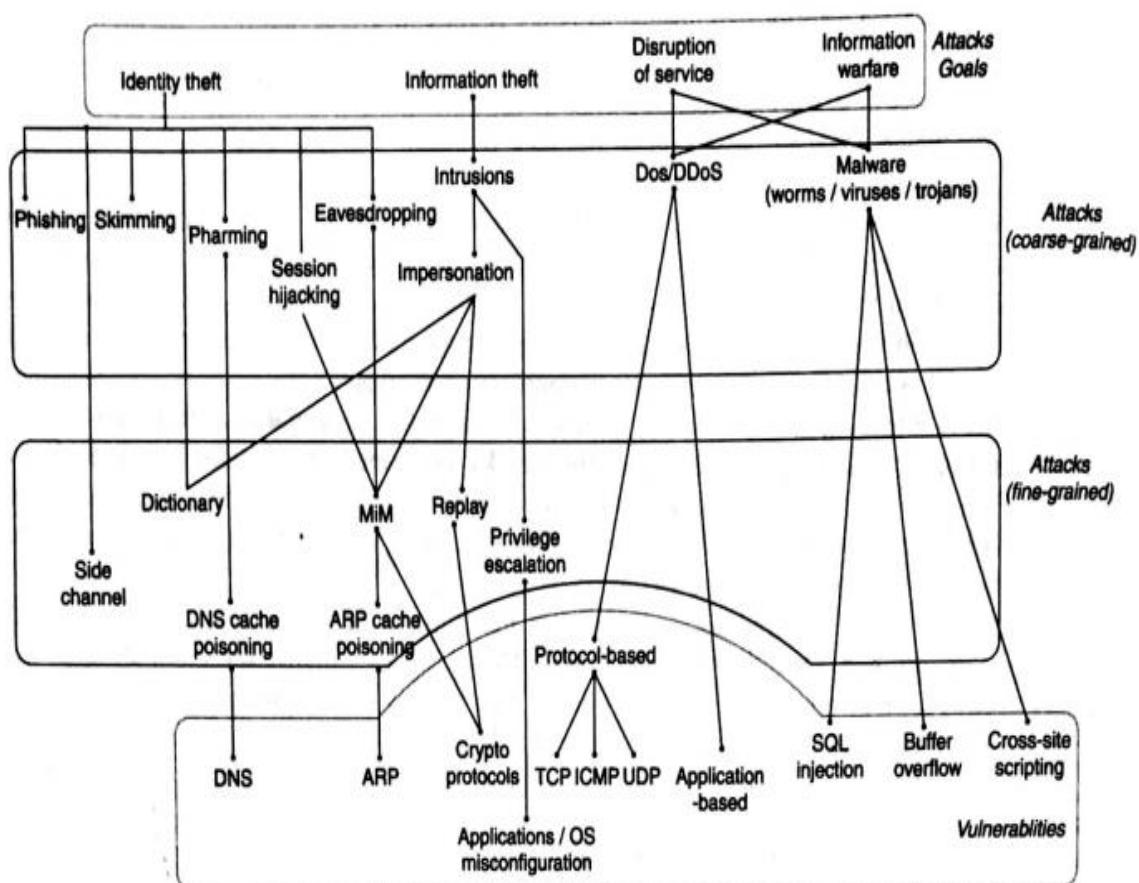


Figure 1.1 common attacks and vulnerabilities

1.1.3 Vulnerabilities

- Behind every attack is a vulnerability of some type or the other.
- Definition: A vulnerability is a **weakness** in a **procedure, protocol, hardware, or software** within an organization that has the potential to cause damage.
- There are at least **four important vulnerability classes in the domain of security:**
 - 1. Human Vulnerabilities:**
 - These are vulnerabilities caused by human behaviour or action.
 - For example, the user clicks on a link in an e-mail message received from a questionable source. By so doing, the user can be directed to a site controlled by the attacker as in a **phishing attack** or a **cross-site scripting attack**.
 - Similarly clicking on an e-mail attachment may open up a document causing a **macro to be executed**.
 - The **macro may be designed to infect other files** on the system and/or spread the infected e-mail to other e-mail addresses harvested from the victim's inbox.

- In both these cases, the **human vulnerability consists of clicking on a link or attachment in an e-mail from a possibly unknown source.**
- The **link or attachment** may have provoked the victim by a flashy message suggesting quick money, etc., blinding him/her to the fact that the message came from an unknown source.
- It is actions like this that make a phishing attack or an e-mail virus so very successful.

2. Protocol Vulnerabilities:

- A number of networking protocols including **TCP, IP, ARP, ICMP, UDP, DNS**, and various protocols used in local area networks (LANs) have features that have been used in unanticipated ways to craft assorted attacks.
- Pharming attacks and various ***hijacking attacks*** are some examples.
- There are tools available ***on-line to facilitate some of these attacks***.
- One such tool subverts the normal functioning of the ARP protocol to sniff passwords from a LAN.
- There are a number of vulnerabilities in the ***design of security protocols*** that lead to ***replay or man-in-the-middle attacks***.
- These attacks, in turn, lead to identity theft, compromise of secret keys, etc.
- Vulnerabilities in network protocols are often related to aspects of their design though they may also be the result of poor implementation or improper deployment.

3. Software Vulnerabilities:

- This family of vulnerabilities is caused by written system or application software.
- In many cases, the causes of the problem seems to be the code that is all too trusting of user input.
- Ex A web server accepts input from a users browser.the web server must accept the request after typing the complete username and password.the server software should perform sufficient validation.

4. Configuration Vulnerabilities:

- These relate to configuration settings on newly installed applications, files, etc.
- ***Read-write-execute*** permissions on files may be too generous and susceptible to abuse.
- The privilege level assigned to a process may be higher than what it should be to carry out a task. This privilege may be misused during some point in its execution leading to what are commonly called "privilege escalation" attacks.
- Besides misconfiguration of software and services, security appliances such as firewalls may be incorrectly or incompletely configured with possibly devastating effect.

1.2 DEFENCE STRATEGIES AND TECHNIQUES

1.2.1 Access Control—Authentication and Authorization

- The first defence strategy to prevent intrusions is access control.
- This implies the existence of a trusted third party that mediates access to a protected system.
- The trusted third party is typically implemented in software and may be a part of the operating system and/or the application.
- The first step in access control is to ***permit or deny entry into the system***.
- This involves some form of authentication — a process whereby the subject or principal (the party attempting to login) establishes that it is indeed ***the entity it claims to be***.
- One form of authentication is the humble password.
- Example: The principal first enters his/her login name. By prompting him/her to enter his/her password, the system implicitly challenges the principal to prove his/her identity.
- In this simple case, knowledge of the secret password constitutes "proof of identity."
- After successful authentication, a subject is logged into the system. The subject may need to access several resources such as files.

1.2.2 Data Protection

- The data in transit or in storage needs to be protected.
- It implies data confidentiality – the data should not be readable by an intruder.
- Another dimension to data protection is the preservation of ***data integrity***.
- This implies that the data while transmitting should not be ***tampered or modified***.
- Cryptographic techniques are among the best known ways to protect both, the confidentiality and integrity of data.
- Cryptography is the science of disguising data and is the subject of the part of this book.
- The encryption operation is performed by the sender which converts the plain text to ciphertext.
- decryption operation is performed by the receiver which converts the ciphertext to plaintext.
- The encryption and decryption operations both use the same secret key known only to the sender and receiver.
- This prevents an eavesdropper from decrypting the encrypted message.
- the computation of the cryptographic checksum uses a secret shared by the sender and receiver.
- The sender computes the checksum as a "one-way function" of the message and secret. It transmits the message and checksum.
- The receiver also computes the checksum. If the computed checksum matches that received, the receiver concludes that there is no error in the received message.

1.2.3 Prevention and Detection

- Access control and message encryption are preventive strategies.
- Authentication keeps intruders out, while authorization limits what can be done by those who have been allowed in.
- Encryption prevents intruders from eavesdropping on messages.
- The cryptographic checksum, on the other hand, detects tampering of messages.
- In the important domain of software security, code testing is used to detect vulnerabilities.
- Blackbox testing is employed when the source code of a program is not available. The goal here is to determine whether the software has been carefully designed to handle unexpected or malicious input.
- For greater assurance of secure software, whitebox testing should be employed. Here, the security engineer has access to source code and can perform more elaborate testing by exercising different control paths in the source code.
- Intrusion preventive techniques can be used to detect anomalous behavior, Continuous monitoring of network logs and operating system logs
- Intrusion detection systems also look for certain patterns of behaviour.
- For example, multiple instances of a given worm often exhibit a characteristic bit pattern called a worm signature.

1.2.4 Response, Recovery, and Forensics

- Once an attack or infection has been detected, response measures should be taken .
- These include shutting down all or part of the system.
- Many intrusion attempts leave information
- Cyber forensics is an emerging discipline with a set of tools that help trace back the perpetrators of cyber crime.
- Table 1.2 defines some of the most widely used terms in cyber security parlance.

Table 1.2 Definitions of commonly used terms in security parlance

- **Security policy** is the set of rules and practices that regulate how an organization manages and protects its computing and communication resources from unauthorized use or misuse.
- A **security mechanism** is a technique or device used to implement a security policy.
- A **vulnerability** is a weakness or flaw in the architecture, implementation, or operational procedures of a system that could be exploited to cause loss or failure.
- Exploitation of a vulnerability with malicious intent leads to a **cyber attack**.
- **Access control** is the process of preventing unauthorized access to a computing or communication resource.
- **Authorization** involves granting a specific entity or process the permission to access restricted data or perform a restricted operation.
- **Auditing** is the process of collecting and analyzing relevant information in order to ensure compliance with security policies laid out for an organization.

One or more of the following are implicit when we talk about a secure connection or session between two parties:

- **Entity authentication** is the process of verifying that the entity being communicated with is indeed the entity it claims to be.
- **Message authentication** is the process of verifying the source or origin of the received message.
- **Confidentiality** is the protection of data from disclosure to an unauthorized party or process.
- **Integrity** is the assurance that data has not been modified, tampered with, or made inconsistent in any way.
- **Non-repudiation** offers a guarantee against repudiation or denial by a party of the fact that it created or sent a particular message.

1.3 GUIDING PRINCIPLES

1. Security is as much (or more) a human problem than a technological problem and must be addressed at different levels.

- At the highest level, security should be addressed by top-level management in large organizations.
- Robust security policies should be formulated and a comprehensive implementation strategy outlined by a dedicated team of security specialists, possibly headed by a Chief Information Security Officer (CISO).
- Some of the mechanisms used to implement high-level policies are in the realm of technology.
- Security engineers have a key role to play in designing techniques and products to protect organizations from the various cyber attacks.
- System administrators handle day-to-day operations.
- They should be proactive in crucial security practices such as patch application.
- One of the key tasks of a system administrator is to configure systems and applications. Their job also involves setting user/group permissions to various system resources such as files, configuring firewalls, sifting through system logs for signs of an intrusion, and processing alerts.
- The final link in the security chain is the rank and file within an organization.
- The employees within an organization should be educated on various do's and don'ts through periodically updated security awareness programs.
- In summary, a healthy combination of enlightened security policy and procedures, backed by enforcement, aided by technology, coupled with diligent

participation of administrators and employees, and presided over by an empowered CISO is the surest insurance against cyber attacks.

2. Security should be factored in at inception, not as an afterthought.

- No one then had thought that those protocols would be abused by attackers in so many creative ways!
- application software (web software, for example) developed today continues to be often vulnerable to numerous attacks such as cross-site scripting and SQL injection attacks.
- The solution lies, at least in part, in integrating secure coding practices into the software curriculum in our colleges and universities.
- In general, security should be factored in early on during the design phase of a new product and then carried forward right through implementation and testing.
- The product could be a networking protocol, a new version of an operating system, a piece of application software, or the architectural layout of computing infrastructure for an enterprise.

3. Security by obscurity (or by complexity) is often bogus.

- There have been a number of cryptographic algorithms proposed which was made mandatory in newly standardized protocols, but their details were not made public.
- The flaws are exposed over time after the protocols have been widely deployed, attracting closer attention from the hacker community.
- There are ethical hackers whose goal is to break software/ protocols/algorithms so that they can be fixed before things get out of hand.
- It is the ethical hacker community at least, if not the public at large, who should be able to study new protocols and algorithms prior to widespread adoption.
- One such example was the procedure followed for selecting an algorithm in the late 1990s for the new secret key cryptography standard — AES was finally chosen after much public scrutiny and debate. As another example, open source software is usually freely available. Public review of its security features can make or break its reputation.

4. Always consider the "Default Deny" policy for adoption in access control.

- The subjects in an access control policy could be people, network packets, operating system processes or even user input.
- One policy is the "Default Permit," i.e., grant the subject's request unless the subject is on a blacklist or it has certain blacklisted attributes.
- The dual of this policy is the "Default Deny" policy. In this case, the subject's request is denied unless it is on a whitelist.
- Clearly, whitelisting is the more conservative approach.

- With whitelisting, the access controller may reject a legitimate subject whose name has been mistakenly excluded from the whitelist but that is the price to be paid for greater security.
- Blacklisting, on the other hand, may accept a bad guy because his name or attributes were mistakenly excluded from the blacklist.
- The tradeoffs between blacklisting and whitelisting should be carefully examined (see Principle 8). However, in general, prudent security design should seriously consider adoption of the "Default Deny" policy.

5. An entity should be given the least amount/level of permissions/privileges to accomplish a given task.

- Role-based access control (RBAC) has influenced a variety of software platforms ranging from operating systems to database management systems.
- The principal idea in RBAC is that the mapping between roles and permissions is paramount.
- The role played by an individual at a given point in time determines the rights or privileges the individual has.
- Conferring higher privilege on an individual than what is warranted by his/her current role could compromise the system.
- Privilege escalation in its different manifestations has caused many security breaches in computer systems.
- The problem often lies in sloppy or incomplete configuration management.
- In publicly accessible servers within an organization such as the web and e-mail servers, unnecessary services hosted by them can open the door to malware, which can compromise those servers. The latter are then used as a springboard to spread to the internal machines in that organization.

6. Use 'Defence in depth' to enhance security of an architectural design.

- This principle is used in many high-security installations and has been recently introduced in some airports. A passenger's ticket is checked before entering the airport terminal building. This is followed by verification of travel documents and inspection of check-in baggage at the airline counter. Next comes a security check (physical) and a further check of the boarding pass, travel documents, and check-in baggage before entering the boarding area (main concourse).
- Defence in depth is applicable to cyber security as well.
- Consider designing the firewall architecture for a mid-to-large size enterprise.
- Every packet from the outside (Internet) should be intercepted by at least two firewalls.
- The firewalls may be from two different vendors and would, preferably, have been configured by two different system administrators.

- They may, and typically do, have some overlapping functionality. Because of differences in the hardware/software design and in configuration, what escapes Firewall 1 may be caught by Firewall 2 and vice versa.

7. Identify vulnerabilities and respond appropriately.

- We have already seen a large number of vulnerability types.
- Vulnerabilities in software or protocols are well researched.
- But equally important are weakness/shortcomings in policy, procedures, and operations.
- How many organizations are geared to implement policies regarding the entry of visitors' laptops and PDAs?
- Or do they even have such policies in place? Such mobile devices and Bluetooth-enabled gadgets may transmit malware to unsuspecting stations within the organization.
- Likewise, USB-enabled PCs may be victims of viruses residing on USB flash drives.
- Often, these organizations have elaborate security infrastructure in the form of firewalls and intrusion detection systems. They securely guard the high-profile main entrance but blissfully ignore the security requirements of the less conspicuous side and rear doors.
- Vulnerability detection and response brings to mind fast-spreading Internet scanning worms.

8. Carefully study the tradeoffs involving security before making any.

- Engineering design often involves making tradeoffs — cost versus performance, functionality versus chip area, etc.
- The previous principle highlighted an important tradeoff — security versus cost.
- Consider, for example, the area of electronic payment involving small purchases (say Rs. 10 or less). Such payments, called micropayments, may be made for digital goods such as on-line news-paper articles.
- Payment schemes use some form of cryptography. The cryptographic overheads of these schemes, in terms of computation cost, can be high. Can we use cheaper (lower overhead) cryptography for micropayments? The downside here is that such cryptography is not as secure. But given the transaction amount, the risk of fraud is probably acceptable.
- In this case, we may be justified in trading off increased security for lower cost. Besides security versus cost, security versus performance is a tradeoff often encountered.

Basics of Cryptography

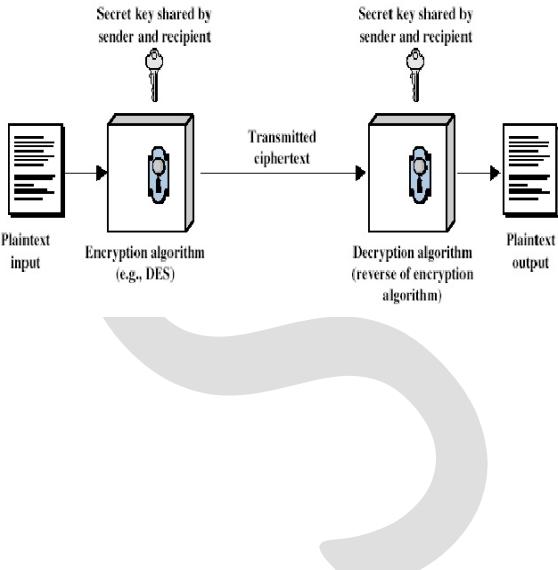
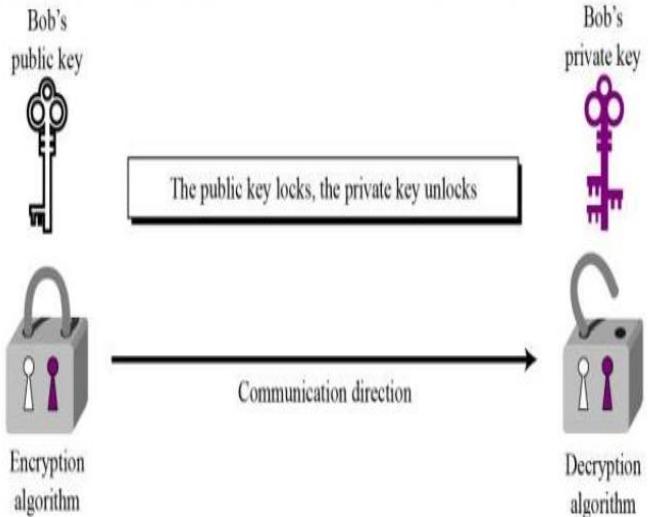
PRELIMINARIES

- Cryptography is the science of disguising messages so that only the intended recipient can decipher the received message.
- Cryptography is the lynchpin of data security — besides providing for message confidentiality, it also helps in providing message integrity, authentication, and digital signatures.
- The original message or document to be transferred is called plaintext
- The plaintext which is encrypted is called ciphertext.
- The process of converting the original plaintext to ciphertext is called encryption
- The process of recovering the original plaintext from the ciphertext is called decryption.
- Encryption involves the use of an encryption function or algorithm, denoted by E, and an encryption key, e.
- Decryption involves the use of a decryption function denoted by D, and a decryption key, d.
- These operations are summarized below.
- $c = E_e(p)$
- $p = D_d(c)$
- Here, p denotes a block of plaintext. It is encrypted by the sender to produce ciphertext denoted by c.
- Decryption operation is performed by the receiver on the ciphertext to recover the plaintext.
- **Kerckhoff's Principle:** The secrecy should be in the key used for decryption, not in the decryption or encryption algorithms.

4.1.1 Secret versus "Public" Key Cryptography

- There are two types of cryptography in widespread use –

1. Secret key cryptography	2. Public key cryptography.
<ul style="list-style-type: none"> ➤ In secret key cryptography, both sender and receiver share a common secret - the same secret key is used for encryption as well as decryption. So $e = d$, this form of cryptography is also referred to as <i>symmetric key cryptography</i>. 	<ul style="list-style-type: none"> ➤ In public key cryptography, two distinct keys forming a key pair are used – <ol style="list-style-type: none"> the encryption key or public key and the decryption key or private key. ➤ The public key of a user(receiver) is used to encrypt messages to that user. ➤ It is the private key of the recipient that is used to decrypt the message. ➤ Because the public and private keys are distinct, this form of cryptography is also referred to as <i>asymmetric key cryptography</i>.
<ul style="list-style-type: none"> ➤ If Alka and Brijesh share a secret key, k, 	<ul style="list-style-type: none"> ➤ Assuming that Brijesh has a public key-private

<p>then she encrypts the message using the common secret.</p> <ul style="list-style-type: none"> ➤ The encrypted message received by Brijesh is decrypted using the same secret. ➤ The secret key operations are summarized below. ➤ Operation performed by Alka ➤ $c = E_k(p)$ ➤ Operation performed by Brijesh: ➤ $p = D_k(c)$ 	<p>key pair, she would encrypt her message using his public key, B_{pu}.</p> <ul style="list-style-type: none"> ➤ Brijesh then decrypts the message using the corresponding private key, B_{pr}. ➤ Assuming that Brijesh keeps his private key securely, he and only he can decrypt the message received from Alka. ➤ The public key-private key operations are summarized below. ➤ Operation performed by Alka: ➤ $c = E_{B_{pu}}(p)$ ➤ Operation performed by Brijesh: ➤ $p=D_{B_{pr}}(c)$
<ul style="list-style-type: none"> ➤ EX: Data Encryption Standard, Advanced Encryption Standard (AES) 	<p>RSA, Elliptic Curve Cryptography (ECC).</p>
<h3>Symmetric Cipher Model</h3>  <pre> graph LR A[Plaintext input] --> B[Encryption algorithm e.g., DES] B -- Secret key shared by sender and recipient --> C[Transmitted ciphertext] C --> D[Decryption algorithm (reverse of encryption algorithm)] D -- Secret key shared by sender and recipient --> E[Plaintext output] </pre>	 <p>Bob's public key</p> <p>Bob's private key</p> <p>The public key locks, the private key unlocks</p> <p>Communication direction</p> <p>Encryption algorithm</p> <p>Decryption algorithm</p>

4.1.2 Types of Attacks

- At a very high level, a cryptographic algorithm is secure if a cryptanalyst (a person with expertise in breaking ciphers) is unable to
 - **(a) obtain the corresponding plaintext from a given ciphertext.**
 - **(b) deduce the secret key or the private key**
- How would the attacker proceed to realize the above objectives? He could accumulate copious amounts of ciphertext.

- He would then look for patterns in the ciphertext in an attempt to reconstruct some plaintext and/or deduce the key. Such an attack which exclusively uses ciphertext is referred to as a "**known ciphertext**" attack.
- Occasionally, all or part of some plaintext blocks are predictable or may be guessed.
- A cryptanalyst may then build a list of corresponding plaintext, ciphertext pairs with the intention of deducing the key. Such an attack is referred to as a "**known plaintext**" attack.
- It may even be possible for a shrewd attacker to carefully choose pieces of plaintext and then induce the sender to encrypt such text.
- An attack on a cryptographic scheme which makes use of pairs of attacker-chosen plaintext and the corresponding ciphertext is referred to as a "**chosen plaintext**" attack.
- The most obvious, though compute-intensive, attack with known plaintext is a **brute force** attempt at obtaining the key by trying all possible key values.
- Let $(p_1, c_1), (p_2, c_2), (p_3, c_3)$ be plaintext—ciphertext pairs.

```

for (each potential key value, k in the key space)
{
    proceed = true;
    i = 1;
    while (proceed == true && i < m)
    {
        if (ci, ≠ Ek( pi)
        {
            proceed = false;
            i ++ ;
        }
        if (i = m+1)
            print (" Key Value is k");
    }
}

```

4.2 ELEMENTARY SUBSTITUTION CIPHERS

4.2.1 Monoalphabetic Ciphers

- The most basic cipher is a substitution cipher.
- For ease of understanding, we consider English text in all the examples in this chapter.
- Let E denote the set of alphabets, (A, B, . . . Z).
- A monoalphabetic substitution cipher defines a permutation of the elements in Σ .
- There are $26!$ permutations; so, there are $26!$ possible monoalphabetic substitution ciphers.
- The simplest substitution cipher is one that replaces each alphabet in a text by the alphabet k positions away (in the modulo 26 sense).
- For $k = 3$, the substitutions are
D for A,

E for B,
A for X,
B for Y, etc.

- Such a scheme is referred to as a **Caesar cipher**.
- A sample plaintext and the corresponding ciphertext for k= 3 is

Plaintext: **WHAT IS THE POPULATION OF MARS**

Ciphertext: **ZKDW LV WKH SRSXODWLRQ RI PDUV**

- In substitution ciphers, like the Caesar cipher, each letter is always substituted for another unique letter. Such ciphers are said to be monoalphabetic.

4.2.2 Polyalphabetic Ciphers

- In a polyalphabetic cipher, the ciphertext corresponding to a particular character in the plaintext is not fixed. It may depend on, for example, its position in the block.
- We next study two examples of such ciphers.

a. The Vigenere Cipher

- The Vigenere cipher is a polyalphabetic cipher that uses a multi-digit key $k_1, k_2, k_3, k_4\dots k_m$
- Here $, k_1, k_2, k_3, k_4\dots k_m$ are each integers.
- The plaintext is split into non-overlapping blocks, each containing **m consecutive characters**.
- Then the first letter of each block is replaced by the letter k_1 positions to its right the second letter of each block is replaced by the letter k_2 positions to its right, and soon.

Plain text	W	I	S	H	I	N	G		Y	O	U	S	U	C	C	E	S	S
---------------	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---

- Key: **04 19 03 22 07 12 05 11 04 19 03 22 07 12 05 11 4**
- Ciphertext: **A B V D P Y L J S N P Q J T X F G V H O Z**
- The first letter in the above text is W. The corresponding key value is 04.
- This means that the ciphertext is the letter 4 positions ahead (in the modulo 26 sense).
- The key length = 8, i.e., the keystring repeats after every **8 characters**.
- There are **four occurrences** of the letter "s" in the above text
- However, each occurrence of "s" is encrypted as a different character in the ciphertext - "V", "X", "O," and "Z".

b. The Hill Cipher

- The Hill cipher is another **polyalphabetic cipher** proposed by **Lester Hill**.
- As in the Vigenere cipher, the plaintext is broken into blocks of size m. However, the key in the Hill cipher is an in $m \times m$ matrix of integers between 0 and 25.

- Unlike the Caesar and Vigenere ciphers, each character in the ciphertext is a function of all the characters in that block.
- Let p_1, p_2, \dots, p_m be the numeric representation of the characters in the plaintext and
- let $c_1, c_2, c_3, \dots, c_m$ represent the corresponding characters in the ciphertext.
- To compute the ciphertext, we map each alphabet to an integer.
- We use the mapping,

A	0
B	1
C	2
D	3
E	4
F	5

- The relationship between a block of plaintext and its ciphertext is expressed by

$$\begin{aligned}
 c_1 &= p_1 k_{11} + p_2 k_{21} + \dots + p_m k_{m1} \bmod 26 \\
 c_2 &= p_1 k_{12} + p_2 k_{22} + \dots + p_m k_{m2} \bmod 26 \\
 &\dots \\
 &\dots \\
 c_m &= p_1 k_{1m} + p_2 k_{2m} + \dots + p_m k_{mm} \bmod 26
 \end{aligned}$$

- This can be conveniently written as
- $C = P K$
- Here, C and P are row vectors corresponding to the plaintext and ciphertext, respectively, and K is the $m \times m$ matrix comprising the key.
- At the receiver end, the plaintext can be recovered from the ciphertext by using
- $P = C K^{-1}$

(refer problem solved in class)

c. One-time Pad

- To perform the one-time pad cipher encryption operation, the pad values are added to numeric values that represent the plaintext that needs to be encrypted.
- Each character of the plaintext is turned into a number and a pad value for that position is added to it.
- The resulting sum for that character is then converted back to a ciphertext letter for transmission.
-

Plaintext: S A C K G A U L S P A R E N O O N E
 Plaintext value: 19 01 03 11 07 01 21 12 19 16 01 18 05 14 15 15 15 14 05
 One-time pad text: F P Q R N S B I E H T Z L A C D G J
 One time pad value: 06 16 17 18 14 19 02 09 05 08 20 26 12 01 03 04 07 10
 Sum of plaintext and pad: 25 17 20 29 21 20 23 21 24 24 21 44 17 15 18 19 21 15
 After modulo Subtraction: 03 18
 Ciphertext: Y Q T C U T W U X X U R Q O R S U O

4.3 ELEMENTARY TRANSPOSITION CIPHERS

- A transposition cipher shuffles, rearranges, or permutes the bits in a block of plaintext.
- Unlike a substitution cipher, the number of 0's and 1's in a block does not change after the shuffling.
- For simplicity, we work with characters (letters) rather than bits.
- Imagine a block of plaintext arranged in a matrix row by row as below.
- **Plaintext: Begin Operation at Noon (any case)**

$$\begin{bmatrix} b & e & g & i \\ n & o & p & e \\ r & a & t & i \\ o & n & a & t \\ n & o & o & n \end{bmatrix}$$

- rearrange the rows as follows
- ROW 1 to row 3
- Row 2 to row 5
- ROW 3 to row 2,
- Row 4 to row 1
- Row 5 row 4.
- The resulting matrix is

$$\begin{bmatrix} o & n & a & t \\ r & a & t & i \\ b & e & g & i \\ n & o & o & n \\ n & o & p & e \end{bmatrix}$$

- now rearrange the columns as follows
- Column 1 to column 4 ,
- Column 2 to column 3,
- Column 3 to column 1,
- Column 4 to column 2.
- The resulting matrix is

➤
$$\begin{bmatrix} a & t & n & o \\ t & i & a & r \\ g & i & e & b \\ o & n & o & n \\ p & e & o & n \end{bmatrix}$$

- The ciphertext thus generated is
- **A T N O T I A R G I E B O N O N P E O N**
- To decrypt the message, the recipient would have to cast the cipher text in a 5×4 matrix, reverse the column shuffles, and then reverse the row shuffles.
- For example, with a combination of guesswork, luck, and limited prior information, a spy might be able to deduce that the planned start time of an attack is 11:15 pm upon receiving the following ciphertext.
- **1 1 K C T A T A M M O C P M 5 1 C E N E**
- This is the ciphertext using the row and column shuffling as in the example above.
- The corresponding plaintext is

Commence Attack 11 15 pm

4.4 OTHER CIPHER PROPERTIES

4.4.1 Confusion and Diffusion

- In 1949, Claude Shannon first proposed the ideas of confusion and diffusion in the operation of a cipher.
- Confusion is the property of a cipher whereby it provides no clue regarding the relationship between the ciphertext and the key.
- Given plaintext p , a sequence of keys k_1, k_2, \dots, k_i and the corresponding ciphertexts are obtained using this encryption $E_{k_1}(p), E_{k_2}(p), E_{k_3}(p), \dots, E_{k_i}(p)$,
- It is nearly impossible to deduce the value of a new, arbitrarily chosen key k_j used to create the ciphertext, $E_{k_j}(p)$.
- Confusion reigns supreme with a cipher if, for any plaintext p if even a single bit in a key k is changed to produce k' , then roughly half the bits in the ciphertexts $E_k(p)$ and $E_{k'}(p)$ are different.
- While confusion is concerned with the relationship between the key and the ciphertext,
- Diffusion is concerned with the relationship between the plaintext and the corresponding ciphertext.

4.4.2 Block Ciphers and Stream Ciphers

Block Ciphers

- With block ciphers, the plaintext is split into fixed size chunks called blocks, and each block is encrypted separately.
- Typically all blocks in the plaintext are encrypted using the same key.
- Block ciphers include DES, AES, RSA, and ECC.
- Block sizes used in secret key cryptography are usually smaller — 64 bits in DES and 128 bits in AES.
- The block size in RSA is much larger — 768 or more bits, while the block size in ECC is about 200 bits.
- If two blocks of plaintext within a message are identical, their corresponding ciphertexts are identical. This statement, however, is only partially true.

Stream cipher

- Stream ciphers typically operate on bits.
- The one-time pad is an example of a stream cipher.
- Practical stream ciphers typically generate a pseudo-random keystream which is a function of a fixed length key and a per-message bit string.
- The key is known to both the sender and the receiver.
- The per-message string could be a message sequence number.
- Alternatively, it could be a random number generated by the sender and transmitted to the receiver along with the encrypted message.
- The ciphertext is itself obtained by performing an \oplus operation between the plaintext and the keystream.
- An example of a stream cipher is RC4 used in the wireless LAN protocol, IEEE 802.11.
- Stream ciphers are usually faster than block ciphers and use less complicated circuits. However, RC4 and some other stream ciphers have been shown to be vulnerable to attack.

Secret key cryptography

5.1 PRODUCT CIPHERS

- Modern day secret-key ciphers are typically synthesized using the Substitution Box (S-Box) and the Permutation Box (P-Box).
- **Substitution Box (S-Box)**
 - An S-box is a device that takes as **input a (binary) string of length m** and **returns a (binary) string 1 of length n**. While it is often the case that $m = n$, this need not always be so.
 - An S-box is implemented using a table (or array) of 2^m rows with each row containing an **n-bit value**.
 - The **input to the S-box** is used to index the table which returns the **n-bit output** of the **S-Box**.
- **Permutation Box (P-Box)**.
 - A P-Box performs a permutation or re-arrangement of the bits in the input.
 - A permutation is more restrictive than a substitution.
 - For example, the number of zeros in the output of the P-Box is equal to the number of zeros in its input while an S-box imposes no such restriction.
 - A P-Box or S-box by itself is not sufficiently powerful to create a secure cipher. However, cascading P-Boxes and S-Boxes alternately, the strength of a cipher can be greatly increased. Such a cipher is referred to as a **product cipher**.
- The three operations that take place in sequence as shown in Fig. 5.1:

(1) An Operation Involving A Function Of The Encryption Key

(2) A Substitution

(3) A Permutation

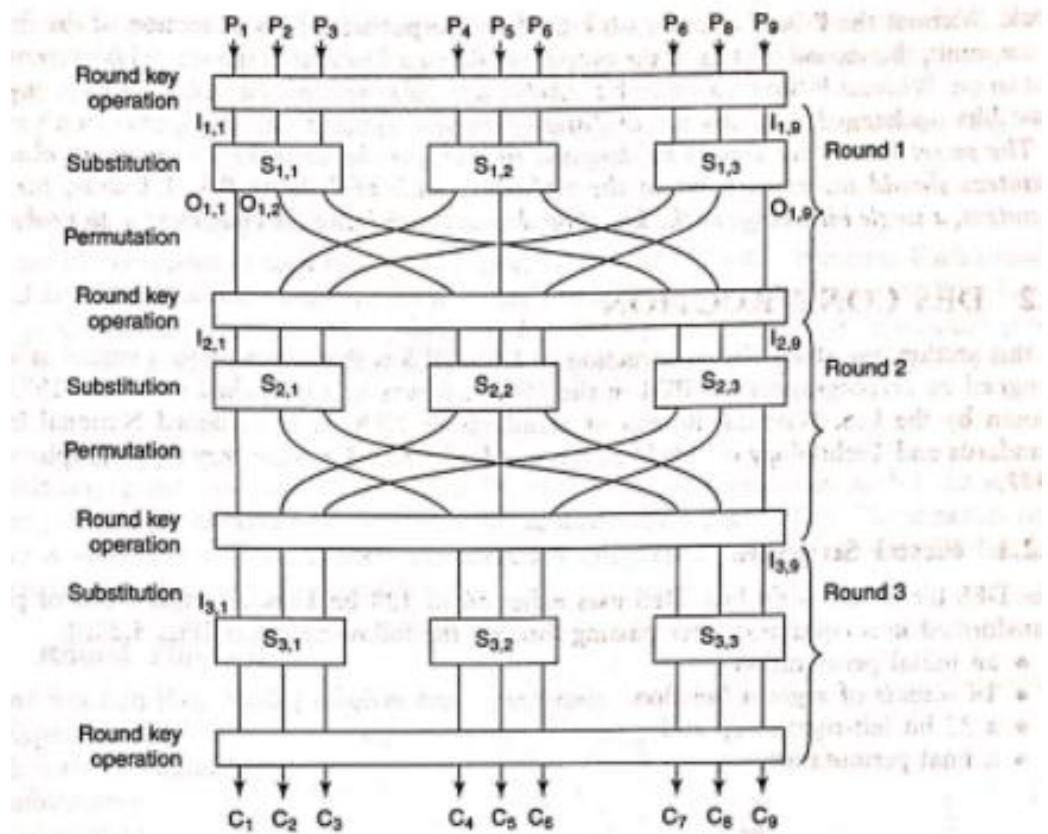


Figure: Three-round SPN network

- These operations are repeated over many rounds or iterations.
- Of the three operations, the first is the only one that involves the **encryption key**.
- It is usually an \oplus (ex or) of the input with the "round" key.
- Each round key is a function of the bits in the encryption key.
- the S-box is usually implemented as a table.
- If the block size of the cipher is b , the size of the table that implements a $b \times b$ S-box is $b \times 2^b$ bits.
- Thus, the table size increases exponentially with the number of inputs.
- As an example, for $b = 64$, the size of the table is 270 bits which is a thousand billion billion bits!
- To save table space, a single S-box is broken into multiple S-boxes as shown in each round of Fig. 5.1.
- If s is the number of S-boxes, the number of inputs to each S-box is b/s .
- Each S-box is now implemented using a table of **size $(b/s)2^{b/s}$ bits**.
- Thus, the total size of all the **S-boxes is $b \times 2^{b/s}$ bits**.

- For a block size of 64, the use of eight S-boxes (each with 8 inputs) would bring down the storage requirements to about 16,000 bits.
- Usage of s box injects non-linearity into the design of the cipher.
- Non-linearity implies the absence of a linear relationship between any subset of bits in the plaintext, cipher text, and key.
- Finally, the third step in each round or iteration is a permutation.
- A P-Box re-orders the inputs that it receives. it diffuses or spreads contiguous bits of the input across the entire block.
- Without the P-Box, the first b/s bits of the output would be a function of the first b/s bits of the input, the second b/s bits of the output would be a function of the second b/s bits of the input and so on.

5.2 DES CONSTRUCTION

- DES is the successor to a cipher called Lucifer designed by cryptographers at IBM in the 1960's.
- It was first published in March 1975 and was chosen by the U.S. National Bureau of Standards or NBS (later re-named National Institute of Standards and Technology or NIST) as the standard cipher for secret key cryptography in January 1977.

5.2.1 Fiestel Structure

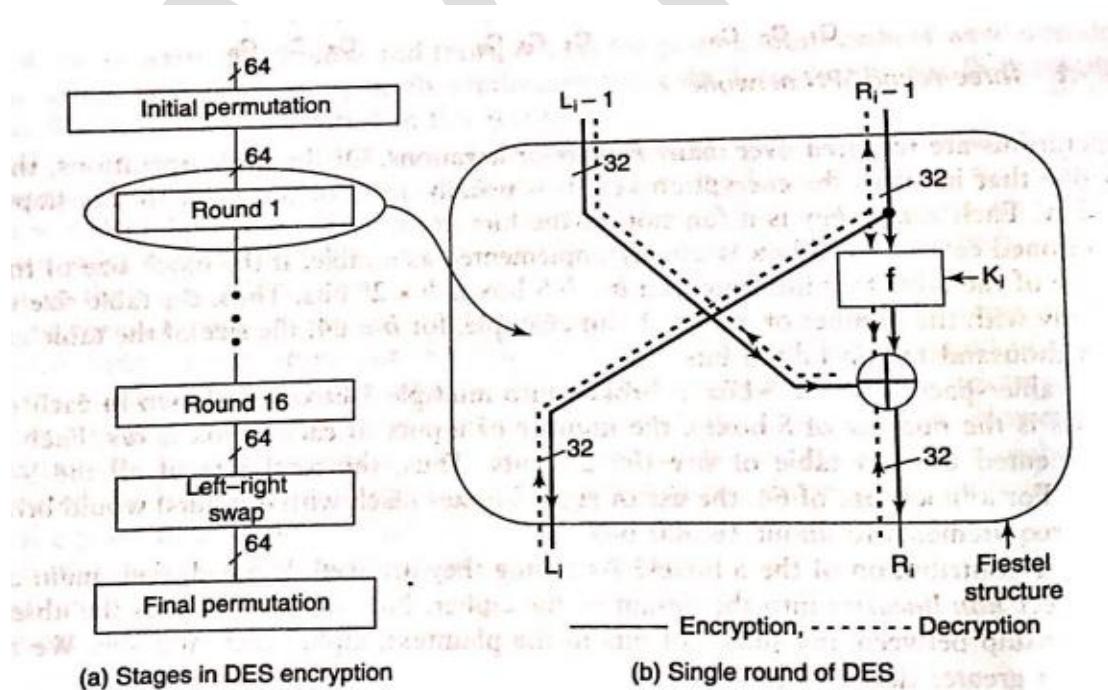


Figure:DES encryption

- The DES data block size is **64 bits**.
- DES uses either **56 or 128 bit keys**.
- A single block of plaintext is transformed into ciphertext after passing through the following stages as shown in above figure:
 1. **An initial permutation**
 2. **16 rounds of a given function**
 3. **a 32-bit left-right swap and**
 4. **a final permutation**
- Each of the 16 rounds is functionally identical.
- The structure of each DES round is explained below.
- Let **L_{i-1} and R_{i-1} be the left and right halves of the input to round i.**
- As shown in above figure:
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- The function f is applied at each round and is referred to as the "**round**" function.
- Each round uses a round key, which is one of the inputs to f.
- Each round key is derived from the DES key.
- The process of decryption involves obtaining L_{i-1} and R_{i-1} from L_i and R_i . Execution proceeds from bottom to top and is summarized by the following equations derived from above Eqs :
- $R_{i-1} = L_i$
- $L_{i-1} = R_i \oplus f(L_i, K_i)$
- The structure of such a cipher is attributed to Horst Feistel (one of the key designers of DES). A cipher that has such a structure is referred to as a **Feistel cipher**.

5.2.2 Round Function

- A round function [above figure (b)] involves four operations:
 - 1) **Expansion**
 - 2) **\oplus with the round key**
 - 3) **Substitution**
 - 4) **Permutation**
- The input to the round function is R_{i-1} , a 32-bit quantity [Fig.(b)].
- This is first expanded into **48 bits** by repeating some bits and interchanging their positions.
- The 48-bit quantity is then **\oplus ed with the round key, K_i** . (which is different for each round).
- The bits in a round key are a function of the bits in the original 56-bit key.
- The result of the \oplus operation is divided into **eight 6-bit chunks**.
- Each chunk is substituted by a **4-bit chunk**
- A total of 8 different S-boxes provide the eight substitutions.
- An S-box is implemented using a **4 x 16 array**.
- Each row of the array is a permutation of the numbers 0 through 15.

- Two bits of the i-th chunk serve as a row index (i_5, i_0) into the i-th table (Fig. 5.3) and the remaining four bits serve as a column index(i_4, i_3, i_2, i_1).
- The output of the S-box is simply the 4-bit string pointed to by the row and column indices.

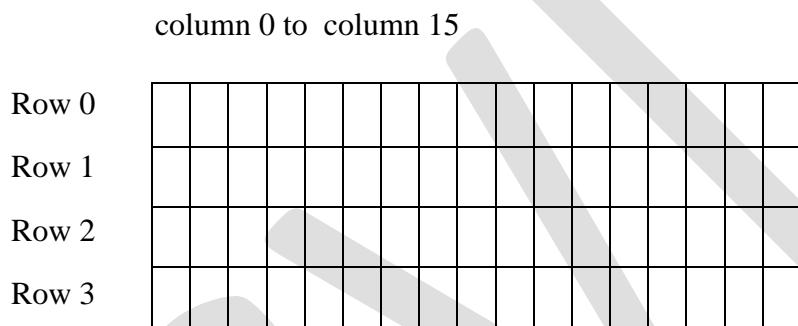
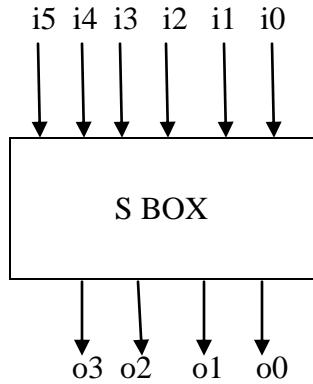


Figure s box implementation using array size 4X16

MATHEMATICAL BACKGROUND FOR CRYPTOGRAPHY.

2.1 Modulo Arithmetic

- Let d be an integer, a dividend.
 m be a positive integer.
 q be a quotient
 r be a remainder.
- The Relationship between d, m, q , and r is

$$d = m * q + r, \text{ where } 0 \leq r < n$$
- $r \equiv d \pmod{n}$
- Let $m=10$ and $r=3$,
 Then $13, 23, 33$ etc all satisfy with quotients $1, 2, 3$ etc.

$$\{-17, -7, 3, 13, 23, 33, 43, \dots\}$$
- Any two numbers in the above set are said to be congruent modulo 10.

→ Set itself is referred to as congruence class.

FACT: If two integers are congruent modulo m , then they differ by an integral multiple of m .

$$a \bmod n = r \rightarrow b \bmod n = r,$$

then

$$a = m * q_1 + r \quad \text{and}$$

$$b = n * q_2 + r$$

$\rightarrow q_1$ and q_2 are integers

By Subtracting,

$$a - b = m(q_1 - q_2)$$

Since q_1 and q_2 are integers,
 a and b differ by an
integral multiple of m .

Modulo arithmetic properties:

1. $(a+b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
2. $(a-b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n$
3. $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$

Example:

Given $m=8$, $a=27$ and $b=34$.

LHS of property 1 from above:

$$\Rightarrow (a+b) \bmod n$$

$$\Rightarrow (27+34) \bmod 8$$

$$\Rightarrow 61 \bmod 8$$

$$\Rightarrow \boxed{5}$$

RHS of property 1:

$$\Rightarrow ((27 \bmod 8) + (34 \bmod 8)) \bmod 8$$

$$= (3+2) \bmod 8$$

$$= \underline{\underline{5}}$$

2.2. THE GREATEST COMMON DIVISOR.

→ Given two integers, a and b , we say a divides b , denoted by $\underline{a|b}$

Definition: If $a|b$ and $a|c$, there exists no $a' > a$, such that $a'|b$ and $a'|c$, then a is referred to as the greatest common divisor of b and c , denoted $a = \gcd(b, c)$.

Example: $\gcd(24, 78) = 6$.

Definition: If $\boxed{\gcd(b, c) = 1}$ we say b and c are relatively prime or co-prime.

EUCLID'S ALGORITHM

- Euclid's algorithm is used to find the gcd of two integers b and c .
- $b > c$
- divide b by c explicitly showing the quotient q and remainder r .
- $b = c * q + r$
- Assign c to b .
 $b = c$.
- Assign r to c .
 $c = r$

Compute $\text{gcd}(161, 112)$

Soln: $b = 161$

$c = 112$

$$b = c * q + r$$

Step 1: $161 = 112 * 1 + 49$

Step 2: $112 = 49 * 2 + 14$

Step 3: $49 = 14 * 3 + 7$

Step 4: $14 = 7 * 2 + 0$

→ Sequence of division continues until a remainder 0(zero) is encountered.

→ Observations about the above procedure:

(a) $\text{gcd}(b, c)$ divides each nonzero remainder above.

↪ $\text{gcd}(b, c) | b$ [$\text{gcd}(b, c)$ divides b]

↪ $\text{gcd}(b, c) | c$ [$\text{gcd}(b, c)$ divides c]

↪ Ex → $\text{gcd}(161, 112) = 7$

7 divides 49, 14.

(b) The remainder just above the zero, under step 4, or the $\gcd(b, c)$.

GCD Theorem: Given two integers b and c , there exist two integers x and y such that

$$b*x + c*y = \gcd(b, c).$$

Corollary 1: If b and c are relatively prime, then there exist integers x and y such that

$$b*x + c*y = 1$$

- In Cryptography, it is often needed to compute multiplicative inverse modulo a prime number.
- The formal procedure to obtain the inverse of c modulo b is called the Extended Euclid's algorithm.
- It assumes b and c are relatively prime, which means $\gcd(b, c) = 1$

Assumptions: $x_1=1$ $y_1=0$
 $x_2=0$ $y_2=1$

Compute

$$x = x_0 - (x_2 * q), \quad b' = b$$

$$y = y_1 - (y_2 * q), \quad c' = c$$

Example:

Q: Compute the inverse of 12 modulo 79.

Soln: Given: $b=79$
 $c=12$.

q	b	c	x	x_1	x_2	x	y_1	y_2	y	$(x * b) + (y * c)$
6	79	12	7	0	1	7	-6	1	-6	$(1 * 79) + (-6 * 12) = 7$
1	12	#	5	1	-1	1	-6	7	-6	$(-1 * 79) + (7 * 12) = 5$
1	#	5	2	1	-1	2	-6	7	-6	$(2 * 79) + (-13 * 12) = 2$
2	5	2	1	1	-1	2	-5	7	-13	$(-5 * 79) + (33 * 12) = 1$

At the end of last iteration,
 $x=1$, and
 $(-5 * 79) + (33 * 12) = 1$

$$33 * 12 = 1 + \underbrace{5 * 79}_{\rightarrow} \equiv 1 \pmod{79}$$

$$\begin{aligned} 396 &= 1 + 395 \\ 396 &= 396. \end{aligned}$$

Thus the inverse of 12 modulo 79 is 33.

Algorithm

```
ComputeInverse(b,c) // Compute the inverse  
{  
    b' = b // Copy b and c to b' & c'  
    c' = c  
    x1 = 1 , y1 = 0 // Assumptions  
    x2 = 0 , y2 = 1  
    r = 2  
    while(r > 1)  
    {  
        q = b/c // Compute quotient  
        r = b % c // Compute Remainder  
        x = x1 - x2 * q // Compute x.  
        x1 = x2 // update x1 and x2  
        x2 = x  
        y = y1 - y2 * q // Compute y  
        y1 = y2 // update y1 and y2  
        y2 = y  
    }  
    b = c // Copy c to b  
    c = r // Copy r to c.  
    return y // last iteration y value.  
}
```

2.3 ALGEBRAIC STRUCTURES.

2.3.1 GROUPS

- A group is the most basic algebraic structure used in Cryptography.
- Definition: A group is a pair $\langle G, * \rangle$, where G is a set and $*$ is a binary operation such that the following hold:

i) Closure: If a and b are elements of G , then $a * b$.

ii) Associativity: If a, b , and c are elements of G then $a * (b * c) = (a * b) * c$.

iii) Identity element: There exist an element I in G , such that for all b in G , $I * b = b = b * I$.

iv) Inverse: For each element b in G , there exist exactly one element c in G such that $b * c = c * b = I$
[c is referred as Inverse of b].

Notation:

Let \mathbb{Z}_n^* denote the set

$$\{i \mid 0 < i \leq n \text{ and } \gcd(i, n) = 1\}$$

i.e., \mathbb{Z}_n^* is the set of all integers modulo n that are relatively prime.

Ex:- $\mathbb{Z}_5^* = \{1, 2, 3, 4\}$ $\gcd(i, n) = 1$.

$$\gcd(1, 5) = 1$$

$$\gcd(2, 5) = 1$$

$$\gcd(3, 5) = 1$$

$$\gcd(4, 5) = 1$$

Ex'- $\mathbb{Z}_6^* = \{1, 5\}$. $\gcd(i, n) = 1$

$$\boxed{\gcd(1, 6) = 1}$$

$$\gcd(2, 6) \neq 1$$

$$\gcd(3, 6) \neq 1$$

$$\gcd(4, 6) \neq 1$$

$$\boxed{\gcd(5, 6) = 1}$$

\downarrow in

$i \leq n$

$5 \leq 6$.

Definitions:

1. The order of a group, $\langle G, * \rangle$ is the number of elements in G .
2. The Euler totient function, denoted by $\phi(n)$, is the order of $\langle \mathbb{Z}_n^*, *_n \rangle$.

Ex: $\phi(5) = 4$

$$\mathbb{Z}_5^* = \{1, 2, 3, 4\}$$

Number of
elements in
 \mathbb{Z}_5^* .

3. Lagrange's theorem: The order of any subgroup, divides the order of the parent group.

→ Consider $\langle \mathbb{Z}_5^*, *_5 \rangle$, its order is 4, no subgroup of it can have order = 3.

→ The subgroups have order 1, 2 and 4.

4. Euler's theorem: If m and n are relatively prime, $m^{\phi(n)} \pmod{n} = 1$.

5. Fermat's little theorem: Let p be prime, and m be a non-zero integer that is not a multiple of p , then

$$m^{p-1} \text{ mod } p = 1$$

6. A group $\langle G, * \rangle$ is cyclic if there is at least one element g in it such that $\langle g \rangle$ is $\langle G, * \rangle$. We refer to such an element of $\langle G, * \rangle$ as a generator of G .

Example :

Check if 2 is a generator of \mathbb{Z}_{13}^* .

$$\mathbb{Z}_{13}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

$\frac{\nearrow \text{if } n}{12 < 13}.$

\rightarrow generator $g = 2$.

\rightarrow $g^{\text{mod } n}$

$$\rightarrow 2^1 \text{ mod } 13 = 2$$

$$2^2 \text{ mod } 13 = 4$$

$$2^3 \text{ mod } 13 = 8$$

$$2^4 \text{ mod } 13 = 3$$

$$2^5 \text{ mod } 13 = 6$$

$$2^6 \text{ mod } 13 = 12$$

$$2^7 \text{ mod } 13 = 11$$

$$2^8 \text{ mod } 13 = 9$$

$$2^9 \text{ mod } 13 = 5$$

$$2^{10} \text{ mod } 13 = 10$$

$$2^{11} \text{ mod } 13 = 7$$

$$2^{12} \text{ mod } 13 = 1.$$

→ $\langle \mathbb{Z}_{13}^*, *_{13} \rangle$ is a cyclic group, which has a generator 2, generates all elements of \mathbb{Z}_{13}^* .

→ hence it is cyclic.

Fact: The group $\langle \mathbb{Z}_p^*, *_p \rangle$ is cyclic, if p is prime.

Fact: Let P be prime, and let P_1, P_2, \dots, P_k be distinct prime factors of $P-1$. Then g is a generator of $\langle \mathbb{Z}_P^*, *_P \rangle$ iff,

$$\frac{g^{(P-1)}/P_i}{\boxed{\frac{g^{(P-1)}/P_i}{\text{mod } P} \neq 1}} \neq 1 \pmod{P} \text{ for all } P_i, \quad 1 \leq i \leq k.$$

The generators of \mathbb{Z}_{13}^* are 2, 6, 11, 7.

Q. Check whether 7 and 3 are generators of $\langle \mathbb{Z}_{13}^*, *_{13} \rangle$.

$$\rightarrow P = 13.$$

$$\rightarrow P-1 = 12$$

Prime factors of 12 are 2 and 3.

$$\begin{array}{r} 2 | 12 \\ 2 | 6 \\ 3 | 3 \end{array}$$

$$\Rightarrow \underline{g^{(P-1)/P_i}} \neq 1 \pmod{P}.$$

Given \rightarrow

$g = 7$
$P = 13$

$$7^{\frac{(12)}{2}} \neq 1 \pmod{P}$$

$$P-1 = 12$$

$$\Rightarrow 7^6 \pmod{P}$$

$$\equiv -1.$$

$$P_i^o = 2 \text{ and } 3$$

$$\Rightarrow \underline{g^{(P-1)/P_i^o}} \pmod{P}$$

$$7^{\frac{12}{3}} \pmod{13}$$

$$7^4 \pmod{13}$$

$$\equiv 9.$$

Hence $\underline{7}$ is a generator of \mathbb{Z}_{13}^* .

$$\Rightarrow g=3, P=13.$$

$$\underline{g^{(P-1)/P_i^o}} \pmod{P}$$

$$3^{\frac{(12)}{2}} \pmod{13}$$

$\equiv 1$. \rightarrow hence $g=3$ is not a generator of \mathbb{Z}_{13}^*

(14)

RINGS.

- A ring is a triplet $\langle R, +, * \rangle$, where $+$ and $*$ are binary operations and R is a set satisfying the following properties.
- $\langle R, + \rangle$ is a commutative group.
- The additive identity is designated as 0 .
- $\langle R, + \rangle$ is a commutative group.
- for all x, y, z in R .
 - 1) $x * y$ is also in R
 - 2) $x * (y * z) = (x * y) * z$. (Associative)
 - 3) $(x * (y + z)) = x * y + x * z = (y + z) * x$.
(Distributive)
- while each element x , has an additive inverse $-x$,
- an element need not have multiplicative inverse (x^{-1}).

Polynomial rings

- Let $\mathbb{Z}_p[x]$ be the set of all polynomials in x with coefficient belonging to \mathbb{Z}_p .
- Addition of two polynomials is addition of coefficients value with modulo p .
- Example Consider two polynomials $a(x)$ and $b(x)$ in $\mathbb{Z}_3[x]$

$$a(x) = 2x^4 + x^3 + 2x + 1$$

$$\begin{array}{r} b(x) = x^5 + x^4 \\ \hline \end{array}$$

$$\begin{array}{r} a(x) + b(x) = (x^5 + 3x^4 + x^3 + 4x + 1) \text{ mod } 3 \\ \hline \end{array}$$

for coeff.

$$1x^5 \equiv 1 \text{ mod } 3 = 1 = x^5$$

$$3x^4 \equiv 0 \text{ mod } 3 = 0 = 0$$

$$1x^3 \equiv 1 \text{ mod } 3 = 1 = x^3$$

$$4x \equiv 1 \text{ mod } 3 = 1 = x$$

$$1 \equiv 1 \text{ mod } 3 = 1 = 1$$

$$\Rightarrow [x^5 + x^3 + x + 1]$$

(16):

Multiplication of two polynomials.
 $a(x) * b(x)$.

$$a(x) = 2x^4 + x^3 + 2x + 1$$

$$b(x) = x^5 + x^4 + 2x$$

$$a(x) * b(x)$$

$$\Rightarrow (2x^4 + x^3 + 2x + 1) * (x^5 + x^4 + 2x)$$

$$\Rightarrow 2x^9 + 2x^8 + 4x^5 + x^8 + x^7 + 2x^4 + 2x^6 + 2x^5 + 4x^2 + x^5 + x^4 + 2x$$

$$\Rightarrow (2x^9 + 3x^8 + x^7 + 2x^6 + 7x^5 + 3x^4 + 2x) \bmod 3$$

$$\Rightarrow 2x^9 + x^7 + 2x^6 + x^5 + 2x$$

$$\Rightarrow \boxed{2x^9 + x^7 + 2x^6 + x^5 + 2x}$$

FIELDS

→ A field, $\langle R, +, * \rangle$ is a commutative ring.

→ R has multiplicative identity 1, additive identity 0.

Chinese Remainder Theorem

- The Chinese Remainder Theorem is used in proving a number of results in cryptography.
- Consider the factorization of an integer, N .

$$N = n_1 * n_2 * \dots * n_k$$

- where n_i and n_j are relatively prime ie $\gcd(n_i, n_j) = 1$.

- $1 \leq i$ and $j \leq k$, $i \neq j$.

- Consider the mapping.

$$f: \underbrace{\mathbb{Z}_n}_{\mathbb{Z}_n} \rightarrow \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \mathbb{Z}_{n_3} \dots \mathbb{Z}_{n_k}$$

$$f(x) = (x \bmod n_1, x \bmod n_2, \dots, x \bmod n_k), \text{ where } x \in \mathbb{Z}_n.$$

- Q. Solve using Chinese Remainder theorem, for $N=30$, Compute $f(x)$.

Solu: $N=30$

→ prime factors for 30 are $\underline{6}$ and $\underline{5}$
 $\uparrow \downarrow$
 $=30$

→ $n_1 = 6, n_2 = 5$

→ $f(x) \rightarrow 0 \leq x < 30$.

$$f(x) = (x \bmod n_1, x \bmod n_2)$$

$$f(0) = (0 \bmod 6, 0 \bmod 5) = (0,0)$$

$$f(1) = (1 \bmod 6, 1 \bmod 5) = (1,1)$$

$$f(2) = (2 \bmod 6, 2 \bmod 5) = (2,2)$$

$$f(3) = (3 \bmod 6, 3 \bmod 5) = (3,3)$$

$$f(4) = (4 \bmod 6, 4 \bmod 5) = (4,4)$$

$$f(5) = (5 \bmod 6, 5 \bmod 5) = (5,0)$$

$$f(6) = (6 \bmod 6, 6 \bmod 5) = (0,1)$$

$$f(7) = (7 \bmod 6, 7 \bmod 5) = (1,2)$$

$$f(8) = (8 \bmod 6, 8 \bmod 5) = (2,3)$$

$$f(9) = (9 \bmod 6, 9 \bmod 5) = (3,4)$$

$$f(10) = (10 \bmod 6, 10 \bmod 5) = (4,0)$$

$$f(11) = (11 \bmod 6, 11 \bmod 5) = (5,1)$$

$$f(12) = (12 \bmod 6, 12 \bmod 5) = (0,2)$$

$$f(13) = (13 \bmod 6, 13 \bmod 5) = (1, 3)$$

$$f(14) = (14 \bmod 6, 14 \bmod 5) = (2, 4)$$

$$f(15) = (15 \bmod 6, 15 \bmod 5) = (3, 0)$$

$$f(16) = (16 \bmod 6, 16 \bmod 5) = (4, 1)$$

$$f(17) = (17 \bmod 6, 17 \bmod 5) = (5, 2)$$

$$f(18) = (18 \bmod 6, 18 \bmod 5) = (0, 3)$$

$$f(19) = (19 \bmod 6, 19 \bmod 5) = (1, 4)$$

$$f(20) = (20 \bmod 6, 20 \bmod 5) = (2, 0)$$

$$f(21) = (21 \bmod 6, 21 \bmod 5) = (3, 1)$$

$$f(22) = (22 \bmod 6, 22 \bmod 5) = (4, 2)$$

$$f(23) = (23 \bmod 6, 23 \bmod 5) = (5, 3)$$

$$f(24) = (24 \bmod 6, 24 \bmod 5) = (0, 4)$$

$$f(25) = (25 \bmod 6, 25 \bmod 5) = (1, 0)$$

$$f(26) = (26 \bmod 6, 26 \bmod 5) = (2, 1)$$

$$f(27) = (27 \bmod 6, 27 \bmod 5) = (3, 2)$$

$$f(28) = (28 \bmod 6, 28 \bmod 5) = (4, 3)$$

$$f(29) = (29 \bmod 6, 29 \bmod 5) = (5, 4)$$

Nagashree. C

Asst Professor, Department of CSE, SVIT

(QD)

- It is straightforward to compute $f(x)$ given x .
- Given a tuple in:
 $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \dots \times \mathbb{Z}_{n_K} \rightarrow \mathbb{Z}_N$.
- Given a tuple,
 $(x_1, x_2, \dots, x_K) \in (\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \mathbb{Z}_{n_3} \times \dots \times \mathbb{Z}_{n_K})$
- $x \in \mathbb{Z}_N$.
- $a_i^o = \frac{N}{n_i}, 1 \leq i \leq K$.
- Let x_i^o denote the inverse of a_i^o in the modulo n_i ,
- $x_i^o \cdot x a_i^o \equiv 1 \pmod{n_i}$.
- x can be computed as:

$$(x_1 x a_1 x x_1 + x_2 x a_2 x x_2 + \dots + x_K x a_K x x_K) \pmod{n_i} = x_i^o$$

for $1 \leq i \leq K$

$$(x_1 \times a_1 \times \alpha_1 + x_2 \times a_2 \times \alpha_2 + \dots + x_k \times a_k \times \alpha_k) \bmod n_i$$

Example. If $N = 210$, $n_1 = 5$, $n_2 = 6$, $\equiv x_i$

Compute $f^{-1}(3, 5, 2)$ ~~$n_3 = 7$~~ ,

Soln: $x_1 = 3 \quad n_1 = 5$
 $x_2 = 5 \quad n_2 = 6$
 $x_3 = 2 \quad n_3 = 7$

$$\rightarrow a_1 = N/n_1 = 210/5 = 42$$

$$a_2 = N/n_2 = 210/6 = 35$$

$$a_3 = N/n_3 = 210/7 = 30$$

$$\rightarrow x_1 = 42^{-1} \pmod{5} = 3$$

$$x_2 = 35^{-1} \pmod{6} = 5$$

$$x_3 = 30^{-1} \pmod{7} = 4$$

$$\begin{aligned} \rightarrow x &= (x_1 \times a_1 \times \alpha_1 + x_2 \times a_2 \times \alpha_2 + x_3 \times a_3 \times \alpha_3) \\ &= (3 \times 42 \times 3 + 5 \times 35 \times 5 + 2 \times 30 \times 4) \end{aligned}$$

$\bmod N$
 $\bmod 210$

$\rightarrow 1493 \pmod{210}$

$\rightarrow \boxed{23}$

Nagashree. C
Asst Professor, Department of CSE, SVIT

(23)