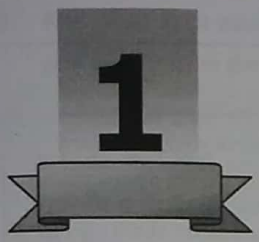

Introduction and Number Theory

Syllabus

At the end of this unit, you should be able to understand and comprehend the following syllabus topics :

- The OSI Model
  - Security Architecture
  - Security Services
  - Security Mechanisms
  - Network Security Model
- Types of Security Attacks
- Classical Encryption techniques
  - Substitution techniques
    - Mono-alphabetic cipher
    - Poly-alphabetic cipher
      - Vignere cipher
      - Playfair cipher
      - Hill cipher
- Transposition techniques
    - Keyed transposition ciphers
    - Keyless transposition ciphers
- Symmetric cipher
- Modular Arithmetic and Number Theory
  - Euclid's algorithm
  - Prime numbers
  - Fermat's and Euler's theorem

1.1 Concept Building - Security - What is it really?

Before we begin with understanding information security and its related concepts, let's talk.

Let me ask you a question	:	"How do you manage your Debit Card and its PIN? Do you leave your Debit Card unattended and with PIN information available to everyone?"
Your Response (Laughingly)	:	"Of course, not. I keep my Debit Card with me all the time and never share my PIN with anyone."
My Response	:	"Oh, that's nice. But, why do you need to do that?"
Your Response	:	"Because, I need to ensure that my money is safe, and no one takes it out except me. I don't trust everyone with my money these days, you know."
My Response	:	"Got it. You are a security champion."

If you followed our conversation, you already know what security is. Our job is easy now. Let us define some terms around our conversation above.

1. **Assets** : You were trying to protect your money, isn't it? It is called Assets. Money is your Asset in our conversation that you were trying to protect.

✍ ***Definition :*** *Assets are something that has value and is worth protecting.*

Security is all about ensuring that the assets are kept protected all the time as much as possible within your capabilities or means.

2. **Controls (or Countermeasures) :** So, how did you actually safeguard your money? You didn't leave the Debit Card around and you memorized your PIN. Isn't it?

✍ ***Definition :*** *Any countermeasures or actions that you take to safeguard an asset are called Controls.*

So, in our *conversation*, you have put two controls in place to safeguard your money – first is to keep your Debit Card with you and second is to memorize your PIN. You are a security champion!

3. **Threat :** Hey, you told me that you don't trust everyone with your money, isn't it? That unknown everyone who can do evil to you or can harm you is called a Threat.

✍ ***Definition :*** *A threat is a person or an entity that can exploit an asset bypassing your controls (if they are weak controls and not enough to safeguard your asset).*

You knew there are threats around your money, and you protected it so well. You are a security champion.

4. **Vulnerability** : What if you left your Debit Card and PIN on the table for anyone to get hold of them and use? I hear you scream, "Come on, why would I do that to myself?". Exactly, You would not want to create a situation in which your assets can be harmed. This is precisely called addressing (or avoiding) a Vulnerability.

✍ ***Definition :*** *Vulnerability is the weakness or lack of controls around assets.*

I am happy that you have put two good controls (keeping your Debit Card safe and memorizing your PIN) and you avoided the vulnerability around your money (asset).

5. **Risk :** So far, you would agree that leaving Debit Card and PIN unattended poses a likelihood that someone might just grab them and use them.

✍ ***Definition :*** *That likelihood of a harm occurring to an asset is called Risk.*

It is this Risk that you want to reduce by applying controls around your assets. Remember one thing here, Risk can NEVER be 0 (zero). Someone can steal your Debit Card from your wallet and force you at gunpoint to tell your PIN.

The core thing that you need to ensure when dealing with Risk is "to reduce it to an acceptable level". Never aim to make anything (or any asset more precisely speaking) risk free because that's not possible, really.

6. **Exposure :** Someday suppose you do accidently leave your Debit Card behind and your PIN was known to someone, you could actually lose some or all your money. That particular day or rather that particular situation of you forgetting your Debit Card behind could lead to an exposure.

✍ ***Definition :*** *Exposure is an instance of being harmed.*

So, if you got exposed anytime, immediately change your PIN and take a lesson in security to apply controls always around assets so that you do not have future exposures. I am sure you won't have exposures because you are a security champion already, aren't you?

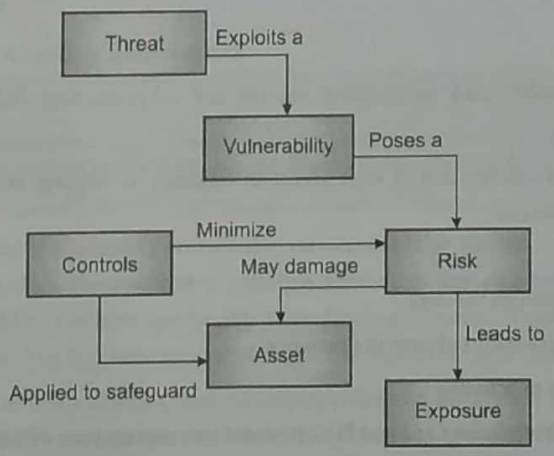

Fig. 1.1.1

Let me summarize the above terms in a simple block diagram.

If you are feeling good about what we talked about so far, believe me, you have started your security journey on a high note. The several chapters and topics that you read in this book (howsoever complex or dry they look at first) are all written to help you effectively do just ONE thing – **Safeguard your Assets**.

If you know what you are ;

- Trying to protect
- And from whom
- And how

You understand security. There is nothing else to learn.

1.2 Concept Building - Information Security Concepts

Q.	Define the goals of security.	MU - Dec. 15, 2 Marks
Q.	Enlist security goals. Discuss their significance.	MU - May 19, 5 Marks

Now that you have a general understanding of security, let's set some context about Information Security. When we say information security – what exactly are we protecting? What is the asset? The asset here is "Information" or more precisely "Digital Information". The information could be about your Facebook user account, Online bank account, OS password, email or pretty much anything that touches a computer system.

There are 3 tenets (or pillars) of security :

1. **C**onfidentiality
2. **I**ntegrity
3. **A**vailability

These tenets in short are also called as the CIA triad or any other combination of the first letters in their words. These are also sometimes called **goals of security**.

Let's dive deeper into each one of them.

1.2.1 Confidentiality

✍ ***Definition :*** *Confidentiality can be defined as, an act of protecting information from unauthorized disclosure to an entity.*

It ensures that the protected information is kept secret throughout its lifetime and is made available only to the authorized entities as and when needed.

The information should be,

1. **Protected at Rest :** When stored on the disk.
2. **Protected in Motion :** When transmitted over the network.
3. **Protected during Use :** When processing.

Remember our conversation from Debit Card and PIN? How did you protect your PIN and provide confidentiality to it?

1. **Protected at Rest :** You didn't write it down. You kept it in your mind. No one could know or use it except you.
2. **Protected in Motion :** You physically moved to an ATM (carrying your mind and the protected PIN there) instead of revealing it to anyone.
3. **Protected during use :** You watch out if someone is looking at your fingers as you punch the PIN on the ATM keyboard.

In terms of digital information, confidentiality is enforced using several mechanisms :

1. Encryption
2. Access control
3. Data classification

We would be studying them at depth in later chapters.

1.2.2 Integrity

✍ ***Definition :*** *Integrity can be defined as, an act of protecting information from unauthorized modification by an entity.*

It ensures that the information remains intact and no unauthorized entity can modify it. Any modification to the information is allowed only if the entity is authorized to do so. The information requires to maintain its integrity throughout its lifetime.

For example, during criminal investigations, any evidence that you collect is protected from touching or any modifications to ensure that those evidences can be used during court proceedings. If an evidence is tampered, it is not admissible in the court and cannot be used. Another example is email. If I send you an email and someone changes it before you read it, you might get wrong information, or it could be severely damaging to our relations.

In terms of digital information, integrity is enforced using several mechanisms :

1. Hashing
2. Access Control
3. Data Classification
4. Input and output sanitization

We would be studying them at depth in later chapters.

1.2.3 Availability

✍ ***Definition :*** *Availability can be defined as, an act of protecting information from unauthorized destruction by an entity.*

It ensures that the information is adequately protected to remain available when it is needed. Any unauthorized entity should not be able to destroy it. Also, the availability principle extends to any equipment such as computers, network devices and printers. These should be available and be able to perform as expected. If someone can get access to them and then prevent you from using these then that impacts availability of the system for your use.

For example, your Windows or Linux systems track all activities done on the system via log files. If I do some mischiefs around your computer and then delete the log files, you would have no way to prove that I did something to your computer. The availability of log files is crucial to ensure that the system is adequately monitored and protected from any security mishaps.

Availability is generally enforced using several mechanisms :

1. Access control
2. Isolation
3. Back up
4. Disaster recovery
5. Business continuity processes

We would be studying them at depth in later chapters.

Let's summarize the above 3 security principles with the help of diagrams.

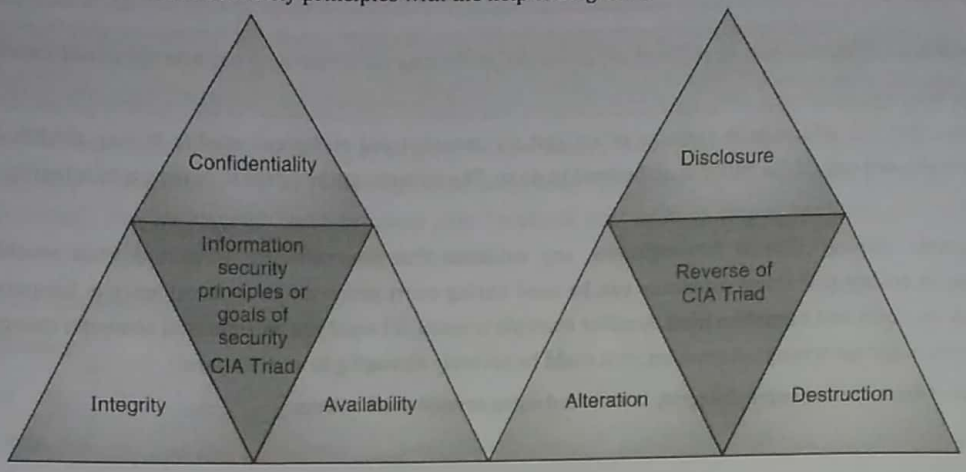

Fig. 1.2.1 : Security Principles

Confidentiality, Integrity and Availability are the 3 core principles of security. Ensuring that you understand the objectives behind these principles is crucial to your success in the information and cybersecurity domain.

1.3 Concept Building - Security Threats and Vulnerabilities

From our previous discussion on Debit Card and PIN, you now understand what security threat and vulnerabilities mean. Threats can exploit your assets and vulnerabilities are situations that could possibly lead to such an exploit.

Let us review some of the security threats and vulnerabilities commonly found in the context of information security.

1.3.1 Security Threats

There are several security threats to an information system. Some of them are briefed as following.

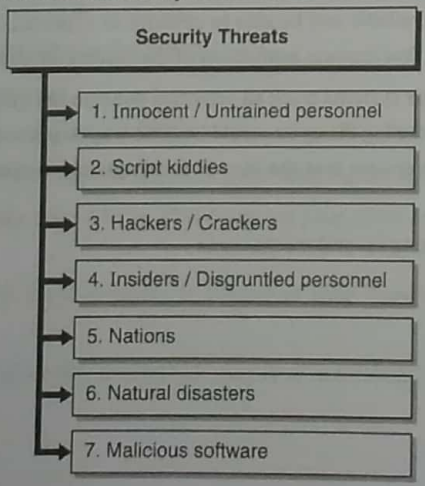

Fig. 1.3.1 : Security Threats

1. **Innocent / Untrained personnel :** These could be your employees, household members or any person who does not understand intricate complexities of security. These people believe the information presented to them and often are soft targets of several frauds. These can get easily convinced and can be pushed to do harm to your organization (say by sharing critical details) or to any other critical asset. As a countermeasure, you should provide security training time to time and enforce the idea that security is everyone's responsibility.
2. **Script kiddies :** These are just exploiting the systems for fun. They have a lot of free time and can go around the internet to find systems that have weak controls. Once a system is found, they can play games, watch movies, download other software or just send some random messages on the screen. These do not have sophisticated skills to exploit weaknesses themselves and usually depend on attack tools or software. As a countermeasure, test your website and software against general attack tools and ensure that any weaker controls are sufficiently addressed.
3. **Hackers / Crackers :** These are people who have sophisticated computer security skills. They have a deep understanding of how various protocols, services, operating systems, drivers, network equipment etc. work and can thus launch sophisticated attacks on such information systems. They usually hide their presence and activities to ensure that they are unnoticed and can exploit the systems for a long time without getting detected. As a countermeasure, invest in penetration testing of your website and software and ensure that all the security findings are adequately addressed.
4. **Insiders / Disgruntled personnel :** These people are on your side, but they have malicious intent to impact your systems. They might have grudge on you or the organization and typically exploit the systems to take revenge. Insider threats are extremely hard to detect since you might believe that their actions are part of their job and may not suspect them or monitor them very closely. As a countermeasure, use access control to provide least possible permissions required to carry out one's job. You should evaluate the permissions time to time and ensure that those permissions are still relevant to the job done by the person.
5. **Nations :** Many a times, nations spy on each other and want to steal information related to country defence, forces, arms, and other intellectual property and confidential information that can severely damage the reputation of the country or its economics. These attacks are highly sophisticated but have huge impact on nations. For example, you might have heard of Russian involvement in the US elections. As a countermeasure, nations typically protect the sensitive information by limiting information sharing only amongst the high-ranking officials. They deploy top notch security solutions, processes and continuously monitor their operations to detect any unauthorized activities.
6. **Natural disasters :** Natural disasters such as flood, earthquake, lightning, etc. can severely damage the information systems (remember availability as one of the tenets of security?) and could impact information availability. As a countermeasure, you invest in backup, business continuity processes and disaster recovery solutions that can quickly bring back the systems and information to avoid large impact on your business.
7. **Malicious software :** These are software programs written with malicious intent. The purpose of these programs is to harm the information systems or extract useful information in an unauthorized manner. As a countermeasure, you install such software detection tools. These tools could be anti-virus, anti-malware, anti-spyware, intrusion detection system, intrusion prevention system, etc. We would explore this in depth in the subsequent section.

1.3.1(A) Comparison between Security Threats

Table 1.3.1 : Comparison between Security Threats

Threat	Skills required	Impact	Detection Possibility
Innocent / Untrained personnel	None	Low - High	Low
Script Kiddies	Medium	Low - High	High
Hackers / Crackers	High	High	Low
Insiders	Low	High	Low
Nations	High	High	Low
Natural Disasters	None	High	Low
Malicious software	Medium	High	High

1.3.2 Security Vulnerabilities

As you recall, vulnerability is the lack of controls around assets. Let us see some of the common security vulnerabilities.

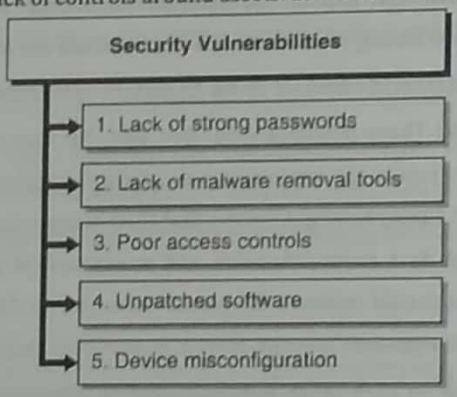

Fig. 1.3.2 : Security Vulnerabilities

1. **Lack of strong passwords :** Passwords are everywhere. For quick and easy recall, you tend to choose simple passwords such as your name, date of birth, school name, etc. People, who know you, are generally aware of this information as well and can try to use this information as your password. Additionally, if you tend to choose simple English words as your passwords, someone can do a dictionary-based attack (more on that later) and can find out your password. As a countermeasure, always choose a complex password that is the combination of uppercase letters, lowercase letters, digits and special characters. Change your passwords regularly and do not use old passwords.
2. **Lack of malware removal tools :** There are several malwares that can be installed on your device as you browse through various websites on the internet. If you do not have a good malware removal tool, overtime, your device might be compromised and impacted. As a countermeasure, install a reputed malware removal tool and update its definitions time to time.
3. **Poor access controls :** Do you allow everyone to be an administrator on your system? Can anyone access anything? If your answers are yes, you probably have not thought the "Least Privilege" principle yet. You should only grant enough permissions as required by the job at hand. No more and no less. Understand the different permissions required and assign them accordingly.

4. **Unpatched software :** Vendors release security patches (software bundles that fix something in the software installed previously) time to time to fix security vulnerabilities found. If you do not install these patches, your system could be prone to exploit because the required security fix to stop the exploit is not installed. As a countermeasure, install software updates as released by the vendor specially the ones that carry security fixes.
5. **Device misconfiguration :** Quite a few times, we configure devices for maximum ease and minimum security. For example, have you locked your phone with a password or a PIN? If you get a security warning, do you click ok even without reading and understanding what the warning is about? When installing an app on your phone, do you ignore to review device permissions that the app would have? Such ignorant behaviour and poor device configuration could weaken the controls that the vendor has put out of factory. As a countermeasure, carefully review your device settings and ensure that they are tuned to provide adequate security.

1.4 Concept building - Access Control and Attacks

Let's understand the basics of access control mechanisms. These mechanisms are at the core of designing security in information systems and are often the target of attacks.

1. Identification

✍ ***Definition :*** *Identification (in short ID) is defined as, a way to claim an entity's presence with respect to the process being carried out.*

This means that during a process, your presence (or your consent) is ascertained (or established). For example, when you try to login to your Facebook account, you provide your Email or Phone number to establish your presence during the login process. There are several other forms of identification that we use today such as Aadhar Card, PAN Card, Voter ID, Debit Card, Admit card, etc. All of these identification methods bring a sense of credibility that you are present, or you give your consent to complete a particular process.

2. Authentication

✍ ***Definition :*** *Authentication is defined as, a way to ensure that the entity is indeed what it claims to be.*

This means that providing just the ID is not enough. You must additionally prove that the ID belongs to you. For example, even if I know your Facebook email address or phone, I cannot login as you until I also know the password. Thus, knowing just the ID is not enough. We need to prove that the ID belongs to us and that is what is precisely called authentication. It is for this reason that you need to additionally sign when you submit Aadhar card or PAN card as an ID proof to ensure that someone didn't just use the photocopy of those IDs without your permission (or consent). Some of the ways to authenticate an ID are passwords, biometric (like your Aadhar fingerprints or phone sensor), PIN (like for Debit Card), or OTP (SMS that you get to confirm transaction).

3. Authorization

✍ ***Definition :*** *Authorization is defined as, a way to determine what resource an entity can access.*

Once you have provided your ID and have been successfully authenticated, the next step is authorization where the system determines if you have the permission to access the desired object. For example, even if you have a valid voter ID card but if your name is not on the electoral list at a particular area booth, you won't be allowed to vote. Having authenticated ID is one thing and getting access to the resource is another. Just because you have an authenticated ID, does not mean that you have automatically access to the resources. So, authenticated ID is a must for authorization but that does not always guarantee that you would be allowed access.

4. Accountability

Definition : *Accountability is defined as, a way to record your actions.*

Suppose, you used a system to take print outs. That system logs this action (pretty much like you record attendance in lab or classroom) to build a trace (evidence or proof) that you used the printer. If you were not supposed to use the printer, the evidence can be used to find you accountable for using it without permissions and could result in particular consequences. Accountability is a key determinant of how securely a system is operating. The logs generated are continuously monitored and necessary alarms are raised if any entry is found to be suspicious.

Let's summarize the 4 access control steps with the help of Fig. 1.4.1.

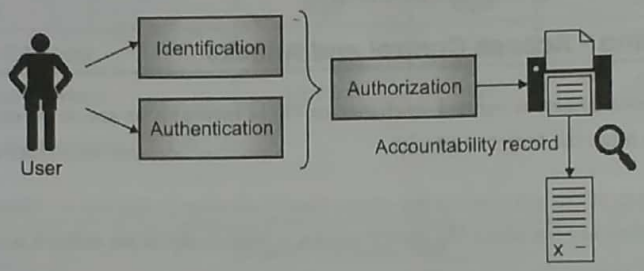

Fig. 1.4.1 : Access Control Steps

5. Non-repudiation

Definition : *Non-Repudiation is defined as, a way to prove your actions.*

It is used in conjunction with accountability and the CIA triad. Non-repudiation provides an assurance that someone cannot deny their actions later on. For example, if I sent you an email, I cannot later deny that I did not. To send an email, I must have used my email ID and password and then sent it over to you over a secure network where no one could change the email body. If you can establish all of these facts truthfully, you have proven that I sent that email and thus established non-repudiation.

Now that you understand the access control steps and security pillars (CIA), let us understand security attacks and what these attacks are actually targeting.

1.4.1 STRIDE Model

STRIDE model is often cited as a reference when designing security for information systems.

Table 1.4.1 : Attacks on Information Systems

Attack Category	Security Property that is attacked
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidentiality
Denial of service	Availability
Elevation of Privilege	Authorization

1. **Spoofing :** When someone tries to steal your identity, it is called spoofing. Spoofing is also called as impersonating someone or something. For example, if I try to login to your Facebook account using your email address and password guesses, I am trying to spoof or impersonate your identity. Spoofing can also be used for fake news, fake websites, fake or malicious files or anything else that could be mistaken for being real. The target of spoofing attacks is to crack authenticity (of person, file, website, news or anything real).
2. **Tampering :** When someone tries to do an unauthorized modification to something, it is called tampering. You would have heard of cricketers tampering with the ball to suit their requirements. It is a punishable offense. Similarly, in information systems, if you try to tamper with say files, emails, or any other information in an unauthorized way, it is called tampering. Basically, you are trying to attack the integrity of the object by making such alterations.
3. **Repudiation :** In this, you are trying to falsely claim that you didn't carry out a particular action. For example, you didn't send an email or didn't visit a website. Remember your childhood days when you broke something, and you try to escape the punishment saying that you didn't actually break it and then your parents finding proofs that it was indeed you who broke it. That's precisely what is non-repudiation. In repudiation attacks, you are trying to destroy evidences that someone can use to falsify your denial claims.
4. **Information Disclosure :** This pertains to unauthorized revealing of any confidential information. In these attacks, the attacker wishes to know the confidential information and tries to crack controls around it to get hold of such confidential information.
5. **Denial of service :** In this type of attacks, the purpose of attack is to make the information system or its services unstable or unavailable to perform its assigned activities. For example, if you can succeed to bring Flipkart website temporarily down, it might mean severe loss of business for Flipkart and the customers may go to a different website for placing their urgent orders.
6. **Elevation of Privileges :** In this category of attack, the attacker tries to get elevated (higher) privileges (permissions / authority) over resources. For example, if you are only a user on a system and you try to attack the system to become an administrator on the system, it is called an elevation of privileges attack.

1.5 OSI Model

Note : Discussing OSI Model in-depth is beyond the scope of this book. It is assumed that you have covered it in detail in your subjects on networking. A general high-level overview is presented here as a refresher.

Definition : *The Open Systems Interconnection Model (OSI Model) is a conceptual model that characterizes and standardizes the communication functions of networked communications without diving into complexities of protocols, architecture and the underlying technologies.*

The OSI model consists of 7 layers. Each layer interacts with the layer above and below it and passes on the respective protocol data units encapsulated into their respective headers.

Table 1.5.1 : Layers of OSI model

Layer Number	Layer Name	Protocol Data Unit	Function
7	Application	Data	Application Interface – APIs, UIs
6	Presentation	Data	Data translation between networking and application
5	Session	Data	Manage communication sessions between sender and receiver

Layer Number	Layer Name	Protocol Data Unit	Function
4	Transport	Segment, Datagram	Reliable data transmission
3	Network	Packet	Network packet addressing and routing
2	Data Link	Frame	Transmission of data between two nodes
1	Physical	Binary	Actual communication over physical media

Note that each OSI layer protocol adds its own information to the data packet.

✍ ***Definition :*** *The process of adding layer specific information and passing on the data to the next layer below is called encapsulation.*

Encapsulation happens top – down (From Application to Physical Layer).

✍ ***Definition :*** *The process of removing layer specific information and passing on the data to the next layer above is called decapsulation.*

Decapsulation happens bottom – up (from Physical to Application Layer).

The Table 1.5.2 shows a quick reference summary for various protocols at the respective OSI Layers.

Table 1.5.2 : Protocols used is Various OSI Layers

Layer Name	Protocols Used
Application	HTTP, FTP, SMTP, etc.
Presentation	JPEG, MPEG, TIFF, ASCII, etc.
Session	NFS, RPC, etc.
Transport	TCP, UDP, SSL, etc.
Network	IP, ICMP, OSPF, etc.
Data Link	ARP, PPP, Ethernet, etc.
Physical	ISDN, DSL, 10Base-T, etc.

1.5.1 The OSI Security Architecture

The objective of the OSI model is to permit the interconnection of heterogeneous computer systems so that useful communication between application processes may be achieved. At the various OSI layers, the security controls must be established in order to protect the information exchanged between the application processes (or the connected computers or devices). Such controls make it difficult to obtain the information in any unauthorized way.

✍ ***Definition :*** *The OSI Security Architecture identifies the basic security services and mechanisms and their appropriate placement at the various layers.*

OSI security functions are concerned only with the OSI layers involved in the communications path. It does not include other security controls such as securing the operating system or the application process itself. Let's learn about the various security services and mechanisms placed at the OSI layers.

1.5.2 Security Services

Q. Define authentication and non repudiation and show with examples how each one can be achieved. **MU - Dec. 16, 5 Marks**

✍ ***Definition :*** *Security Services are safeguard controls recommended to be placed at the various OSI layers.*

The various security services are listed as shown in Fig. 1.5.1.

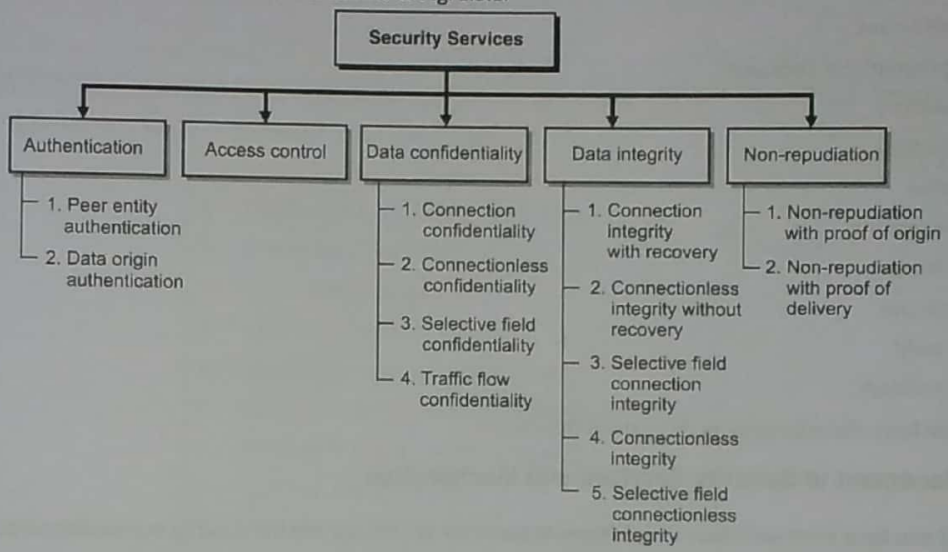

Fig. 1.5.1 : Security Services

1.5.3 Security Mechanisms

Q. Specify mechanisms to achieve each goal. **MU - Dec. 15, 3 Marks**

Q. List with examples the different mechanisms to achieve security. **MU - May 16, 5 Marks**

✍ ***Definition :*** *Security mechanisms are various techniques recommended to provide security services at the various OSI layers.*

The various security mechanisms that can be applied are as following.

- Encipherment (Encryption)
  - Symmetric
  - Asymmetric
- Digital Signature
  - Signing a data unit
  - Verifying a data unit
- Access control
  - Passwords
  - Time of access
  - Duration of access
  - Access route

- Data integrity
  - Sent quantity of data
  - Received quantity of data
  - Sequencing of data units
  - Time stamping
- Authentication
  - Handshaking
  - Cryptographic techniques
- Traffic padding
- Routing control
- Notarization
- Pervasive security
- Security labels
- Event detection
- Security audit
- Security recovery

You would learn these techniques throughout this course.

1.5.4 Placement of Security Services and Mechanisms

Now that you have a fair understanding of the various security services and the security mechanisms that can be used at the various OSI layers, let us see recommended placement for them.

Table 1.5.3 : Layerwise Security Services and Mechanisms

OSI Layer	Security Service	Security Mechanism
Physical Layer	• Connection Confidentiality • Traffic Flow Confidentiality	Encipherment
Data Link Layer	• Connection Confidentiality • Connectionless Confidentiality	Encipherment
Network Layer	• Peer Entity Authentication • Data Origin Authentication • Access Control service • Connection Confidentiality • Connectionless Confidentiality • Traffic Flow Confidentiality • Connection Integrity without recovery • Connectionless Integrity	Authentication Encipherment Digital Signature Access control Routing control Traffic Padding Data integrity
Transport Layer	• Peer Entity Authentication • Data Origin Authentication	Authentication Encipherment

OSI Layer	Security Service	Security Mechanism
	• Access Control service • Connection Confidentiality • Connectionless Confidentiality • Connection Integrity with recovery • Connection Integrity without recovery • Connectionless Integrity	Digital Signature Access control Data integrity
Session Layer	No security services are provided in the session layer	Not Applicable
Presentation Layer	• Connection Confidentiality • Connectionless Confidentiality • Selective Field Confidentiality • Traffic Flow Confidentiality • Peer Entity Authentication • Data Origin Authentication • Connection Integrity with Recovery • Connection Integrity without Recovery • Selective Field Connection Integrity • Connectionless Integrity • Selective Field Connectionless Integrity • Non-repudiation with Proof or Origin • Non-repudiation with Proof of Delivery	Encipherment Digital Signature Data integrity Notarization
Application Layer	• Peer Entity Authentication • Data Origin Authentication • Access Control Service • Connection Confidentiality • Connectionless Confidentiality • Selective Field Confidentiality • Traffic Flow Confidentiality • Connection Integrity with Recovery • Connection Integrity without Recovery • Selective Field Connection Integrity • Connectionless Integrity • Selective Field Connectionless Integrity • Non-repudiation with Proof of Origin • Non-repudiation with Proof of Delivery	Encipherment Access Control Digital Signature Data integrity Traffic Padding Notarization

1.6 Network Security Model

Definition : *Network Security Model (NSM) is a seven-layer model that divides the task of securing a network infrastructure into seven manageable sections.*

It is similar to the seven OSI layers. The model is generic and can apply to all security implementation and devices. NSM provides a unified way of securing networks. It is easier to pinpoint issues at the respective NSM layers and address the gaps, if any.

The Table 1.6.1 lists the NSM layers and how they align with the OSI layers. It is important to understand that like the OSI layers, each NSM layer builds on top of the previous layer. If any layer is compromised, the layers above it are disrupted as well. For example, if there is an attack at NSM layer 2, it would disrupt layers above it (3, 4, 5, 6 and 7). Let's learn about each of the NSM layers.

Table 1.6.1 : Layers in Network Security Model

Network Security Model (NSM)	OSI Model (inverted)
Physical	Physical
VLAN	Data Link
ACL	Network
Software	Transport
User	Session
Administrative	Presentation
IT Department	Application

1. **NSM Layer 1 – Physical :** It works at the physical layer. It ensures to safeguard the physical aspects of network. For example, physical access to the routers, switches or any other networking equipment. There could be several physical forms of physical security such as security alarms, security guards and CCTV.
2. **NSM Layer 2 – VLAN :** VLAN stands for Virtual Local Area Network. At this layer, the network is segmented (partitioned) into smaller network chunks to safeguard them individually and to also manage them effectively. It ensures that only authorized devices connect to the provided networks. You could create VLANs department wise, region wise or in any other suitable grouping mechanism based on your site requirements.
3. **NSM Layer 3 – ACL :** ACL stands for Access Control List. ACLs are created to allow or deny access based on the network layer from the OSI layer. For example, certain IP ranges (say finance department) might be restricted for access by other devices on the network. ACLs can be created on routers, firewalls and switches and can effectively control the network access as designed and intended.
4. **NSM Layer 4 – Software :** The software layer is focused on keeping the device software up to date with the latest upgrades and patches in order to mitigate any known software vulnerabilities. At this layer, the patches are installed to ensure that the software running on the device cannot be exploited. For example, you install security patches on your operating system or update applications on your phone to ensure that you are running the secure version of the software and it does not have any known exploits.
5. **NSM Layer 5 – User :** This layer deals with the user access and management. The user layer focuses on the user's training and knowledge about security on the network. The user should understand the basic concepts of network security and should be capable of applying security related judgement. For example, users should be aware of which software to run on the system and which not.

6. **NSM Layer 6 – Administrative :** The administrative layer focuses on the training of administrative users. It works very similar to the user layer but focuses primarily on the administrative staff. It provides guidance to the layers below it to adequately protect the network. For example, it can dictate which software is allowed for user consumption.
7. **NSM Layer 7 – IT Department :** The IT department layer deals directly with the maintenance of all layers and making sure that the entire network works correctly from NSM model. It has several professionals in the team that know to architect and operate a secure network.

1.7 Types of Security Attacks

Q. List and explain various types of attacks on encrypted message. **MU - May 18, 5 Marks**

At a high level, there are two broad categories of security attacks carried over the network – active attacks and passive attacks are shown in Fig. 1.7.1.

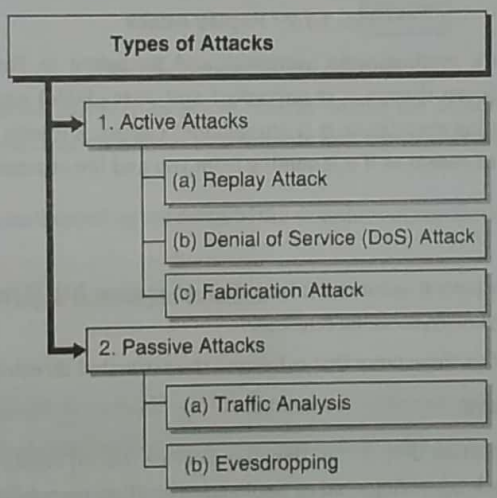

Fig. 1.7.1 : Types of Security Attacks

1.7.1 Active Attacks

✍ ***Definition :*** *An Active attack is defined as, an attack where the attacker actively participates in the communication or the attack mechanism and disrupts the systems by sending several manipulated inputs.*

In a nutshell, the attacker intercepts (captures) the communication channel, and manipulates the communication going over it. Another variation of the active attack is when the attacker continuously disrupts the ability of the system to process the information correctly. Let us expand on some of the examples of active attacks.

1. Replay Attack

- This is like replaying a song. The attacker captures the real and specific communication packets, stores them with herself, and then sends it at a later point in time as though she is sending the information for the first time like an authentic information.

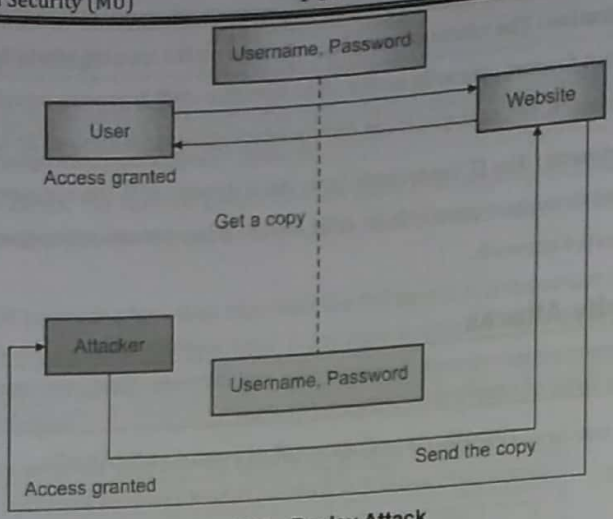

Fig. 1.7.2 : Replay Attack

- **For example :** Suppose you are sending your username and password to Facebook. Someone can capture that information over the network (even though it is encrypted and unreadable) and without even knowing the actual content of the captured information ever (since it is unreadable) can store it with herself. At a later point in time, the same captured information can be resent as if it is coming from you and the attacker might get access to your Facebook account.

Countermeasures

- You can use timestamps and sequence numbers (also called as session ID). If the message comes with a sequence number that is already used previously, it can be rejected.
- Similarly, if a message comes with a timestamp that is beyond the estimated threshold, it can be rejected.

2. Denial of Service (DoS) Attack

- Denial of Service (DoS) or its variation Distributed Denial of Service (DDoS) refers to a category of attacks that can be aimed at various layers of network, operating system, application or other parts of information systems.

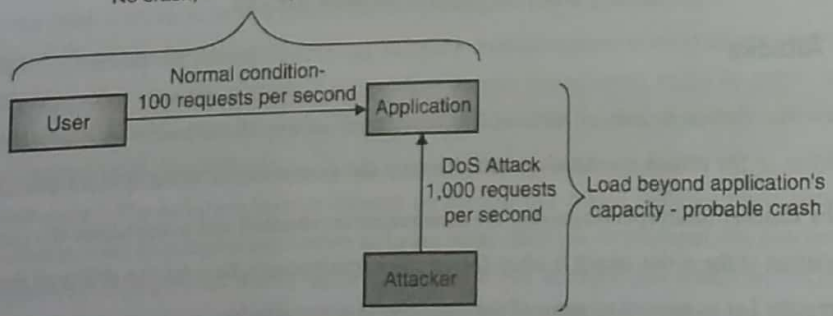

Fig. 1.7.3 : Denial of Service (DoS) Attack

- In this type of attack, the attacker overloads the system beyond its capacity such that the system or controls around it fail. Once the system fails, it is no more available for performing its assigned activities.
- **For example :** An application might be capable of processing a maximum of 100 requests per second. Attacker would typically send over 1,000 requests per second such that the application fails to cope up with it and crashes. The sole motivation behind DoS attacks is to bring down the availability of the system.

Countermeasures

- Some of the countermeasures to protect from DoS are firewall, application limit, white listing networks, etc.
- Firewall can be used to drop network connections that come from a particular location or based on other networking parameters (a list of allowed IP addresses, etc.).
- Application limits can protect application from crashing when the rate of requests goes beyond a set limit.

3. Fabrication Attacks

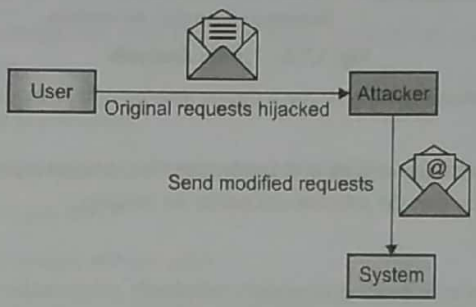

Fig. 1.7.4 : Fabrication Attacks

- Fabrication attack is again a broad category of active attacks where the attacker deliberately modifies messages, parameters, properties, etc. of information system components and try to alter the behaviour of the system often by passing security controls.
- SQL injection, masquerade and email spoofing are some of the examples of fabrication attacks.

Countermeasure :

- Hashing, redundancy checks, and input and output validation.

1.7.2 Passive Attacks

✍ ***Definition :*** *A passive attack is defined as, an attack where the attacker does not alter the behaviour of the information system and silently performs her malicious activities.*

Unlike active attacks, passive attacks are predominantly used to learn information such as number of systems, how the system operates and behaves under various circumstances and general operational characteristics such as the name and version of the operating system running on the targeted machines. Once the information is gathered, the attacker launches complex and more impactful attacks. Let us expand on some of the examples of passive attacks.

1. Traffic Analysis

- In traffic analysis, the network traffic and its patterns are watched out over a period of time to infer important information and guess possible activities.
- **For example :** Following is some information that can be possibly guessed just by knowing the traffic volume and patterns.
  - Long communication : It can denote some emergency.
  - Short communications : It can denote planning, checking, and negotiation.
  - No communication : It can indicate a lack of activity.
  - Time of communication : It can indicate who works when.

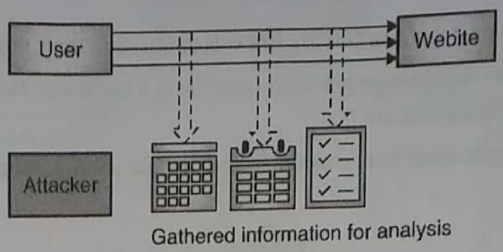

Fig. 1.7.5 : Traffic Analysis

- Such an information gathering exercise can give you a decent amount of information about your target to launch more sophisticated attacks.
- Some of the countermeasures to traffic analysis is to randomize the communication or send fake traffic time to time to degrade the quality of information that the attacker can gather for analysis.

2. Eavesdropping

- Eavesdropping is very similar to over hearing someone's telephonic conversation taking advantage of your proximity to the person.
- You can be standing close enough to the person to hear what she is speaking over phone or also hear what the other party on the phone is saying, especially in calm areas. Similarly, in digital communication, you can wire tap and get a copy of information being communicated and spy on it.
- Spying on such communication by tapping the information as it is being sent or received is called eavesdropping.

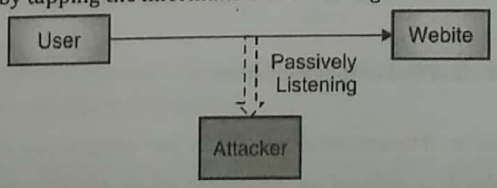

Fig. 1.7.6 : Eavesdropping

- Some of the countermeasures to eavesdropping is sending noise time to time or using random channels of communication.

1.7.3 Comparison between Active and Passive Attacks

Table 1.7.1 : Comparison between Active and Passive Attacks

Comparison Attribute	Active Attack	Passive Attack
Complexity	High	Low
Impact	High	Low
Detection Possibility	High	Low
Prevention Possibility	High	Low
Duration of attack	Short	Long
System Behaviour	Modified	Unaffected
Original Information	Modified	Unaffected
Purpose	Harm the ecosystem	Learn about the ecosystem

1.8 Concept Building – Information Secrecy

Consider a scenario. You and your friend are in different cities and are chatting over phone using an application. You are discussing a new idea to found a startup and have some cool plans that can change the world and make the business very profitable.

But wait, I have two questions for you;

1. How do you know that your conversation (you and your friend sending and receiving text from each other) is not available for anyone else to read?

   [**Goal :** Information not available to everyone].

2. Don't you think that your business plan is confidential and the conversation between you and your friend should somehow remain readable only to you both?

   [**Goal :** Information available to intended entities only].

If you realize the importance of protecting the digital confidential information, you so much wish that there was a way in which you could ensure that the information is available only to you both – blink your eyes and wish granted – Welcome to the world of Cryptography!!

The dictionary meaning of cryptography is "secret writing".

Digitally speaking,

✍ ***Definition :*** *Cryptography is a science and a method of storing and transmitting information in a form that only those it is intended for can read and process its.*

In a nutshell, the information is available in a readable form only to those who are authorized.

Let's take a different example to understand a related concept. You might have used coupon codes in online shopping. Based on the offers running on the website, the coupon code (a set of characters) discounts the price on your chosen items. You apply the coupon code to your cart and based on various terms and conditions, the discount amount is calculated. Sometimes, you get a flat off and sometimes a percentage of the cart value.

In this scenario, we have two interesting concepts;

- The coupon code (a unique set of alphabets and digits).
- The coupon code processing terms and conditions (the algorithm (rules) that determines how to apply the coupon code and how much discount to actually give you).
- In a different example, you and your friend might have some words (or codes) that are only understood by you both. When you use that word, your friend knows what exactly that really means even if you say it loud and clear and others hear it.

For example : Suppose you both have decided that when you say, "I eat banana", that would actually mean "Let's bunk college today". Now, when you say it, your friend gets the real meaning whereas everyone else thinks that you were referring to a fruit.

So, an important concept to understand here is that when the real information is hidden within what is being communicated, the communicated information can be stored or transmitted without disclosing the actual meaning and we don't care if someone gets the communicated information because the real information is hidden. So, real information can freely move around hiding within the communicated information.

1.9 Concept Building - Introduction to Cryptography

Now that you understand some of the scenarios around information secrecy, let's define some the basic terms that we would be using throughout the chapter.

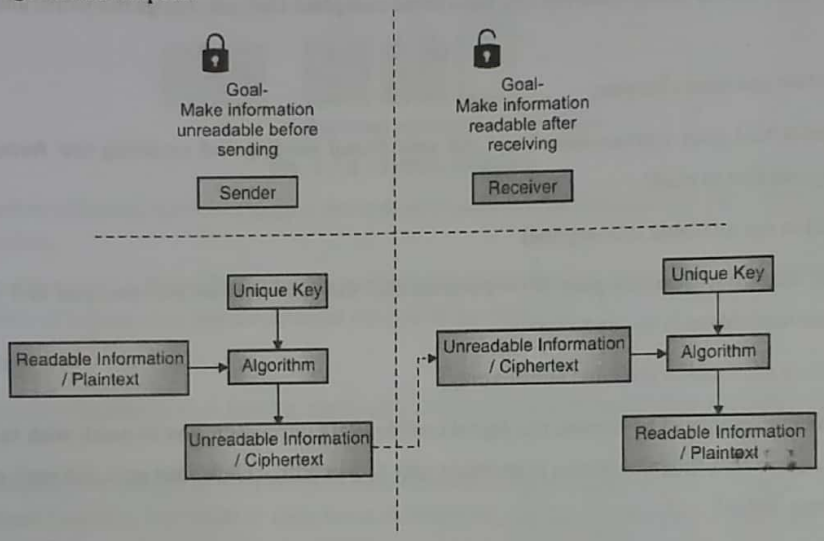

Fig. 1.9.1 : How Cryptography Works

1. **Goal of cryptography :** As you understand, the core goal of cryptography is to hide confidential information. So, cryptography majorly provides Confidentiality out of the CIA triad (Confidentiality, Integrity and Availability).
2. **Information/ Data :** This is the asset that is being protected (provide confidentiality using cryptography in this case). An asset is something that has value and is worth protecting. The sender ensures that the information is only readable by the intended receiver even if it is captured/ seen by anyone else.
3. **Unique Key :** This is a set of numbers (like your coupon code) that helps to make the information secret. It is like your usual key that helps to lock and unlock the door. Note here that like various bikes could have the same locking mechanism, a key is unique to a bike even if the bikes are of same model.
4. **Algorithm :** Algorithm is a process or set of rules to be followed to make the information secret. There are various algorithms (like various types of locks) available that make the information secret in different ways using the keys. In cryptography, such algorithms are also called ciphers.
5. **Plaintext :** The information that is readable and understandable is called plaintext.
6. **Ciphertext :** The information that is not-understandable even if you can read it is called ciphertext.
7. **Encryption (or encipher) :** Encryption is a method of converting plaintext into ciphertext by using an algorithm and a key.

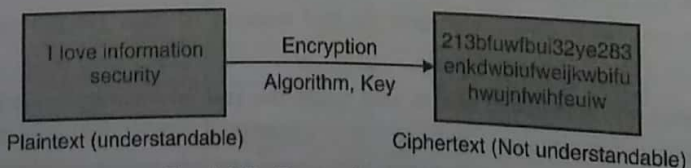

Fig. 1.9.2 : Encryption (or encipher)

8. **Decryption (or decipher) :** Decryption is a method of converting ciphertext into plaintext by using an algorithm and a key.

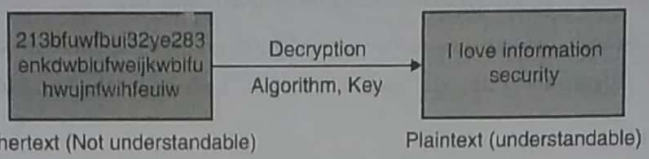

Fig. 1.9.3 : Decryption (or decipher)

Crypto Entities	Is Secret?	Is understandable?
Plaintext	No (But, need to)	Yes
Ciphertext	No	No
Key	Yes	No
Algorithm	No	Yes

Note : The rest of the sections in this chapter heavily build upon the concepts and the introduction you got. Please take some time to make yourself familiar and comfortable with the terms before reading further. Classical Encryption Techniques.

1.10 Classical Encryption Techniques

Q. What are traditional ciphers? **MU - May 19, 2 Marks**

Encryption typically involves two operations as shown in Fig. 1.10.1.

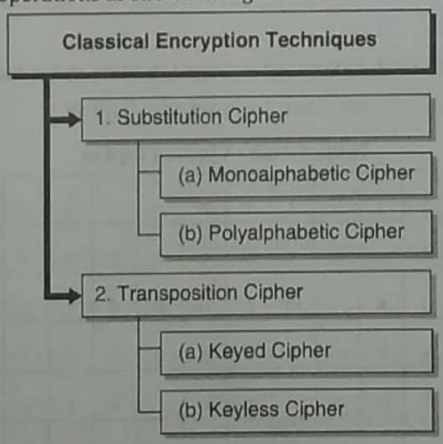

Fig. 1.10.1 : Classical Encryption Techniques

1.10.1 Substitution Cipher

Q. Define the following with example :
(i) Substitution cipher (ii) Polyalphabetic cipher **MU - Dec. 15, 4 Marks**

Q. With the help of examples compare and contrast mono-alphabetic ciphers and poly-alphabetic ciphers? **MU - Dec. 17, 5 Marks**

Q. Discuss any one substitution cipher with example. List their merits and demerits. **MU - May 19, 4 Marks**

- In this operation, one character is replaced by another (like substitutes in games). For example, A can be substituted by D and B can be substituted by E and so on based on a chosen substitution key.
- Let's take an example. Assume that our substitution key is "shift next by 3". Recall from earlier discussion that the key is preserved secretly. Our algorithm (rule) is simple substitution. Let's apply the key and algorithm and encrypt some plaintext.

Table 1.10.1 : Simple Substitution Table

Plaintext	Ciphertext
I love cybersecurity	L oryhfbehuvhfxulwb
Apple	Dssoh
23456	56789

- The Table 1.10.1 is a simple substitution table where each character in plaintext is moved by 3 characters. Characters such as "y" when shifted, take the form of y→z, z→ a, a→b.
- The above example is a classical substitution cipher called **Caesar cipher** named after Julius Caesar. This type of substitution cipher is also referred to as a **"monoalphabetic substitution cipher"** because it uses only one character at a time.
- Another type of substitution cipher is called **"polyalphabetic substitution cipher".** In this, more than one alphabet is used at a time for encrypting plaintext. Let's learn a few polyalphabetic substitution ciphers.

1.10.1(A) Vignere Cipher

Table 1.10.2 is the Vignere table where alphabets are arranged in rows and columns. It is simple to draw. Just write a-z skipping one alphabet from left, at a time, in a row. First two rows are identical.

Table 1.10.2 : Vignere Cipher

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j

l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Let's take an example:

Plaintext : Apple

Security Key : Snow

Algorithm : Vignere Cipher

Let's try to find out ciphertext given the information above. You will need to superimpose the security key above the plaintext you wish to encrypt. You would find the ciphertext value for each plaintext character at the intersection of column (from security key value) and row (from plaintext value).

Security Key →
Plaintext →

s	n	o	w	s	n	o	w	← column
a	p	p	l	e				← row

Ciphertext would be :

- First letter (ciphertext of plaintext a) → intersection of column s and row a →value is s
- Second letter (ciphertext of plaintext p) → intersection of column n and row p →value is c
- Third letter (ciphertext of plaintext p) → intersection of column o and row p → value is d
- Fourth letter (ciphertext of plaintext l) → intersection of column w and row l → value is h
- Fifth letter (ciphertext of plaintext e) → intersection of column s and row e → value is w

Table 1.10.3 : Vignere Cipher Example

Plaintext \ Security Key	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
b	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
c	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
d	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
e	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
f	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
g	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
h	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
i	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
j	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
k	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
l	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
m	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
n	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
o	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
p	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
r	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
s	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
t	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
u	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
v	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
w	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
x	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

So, the ciphertext for plaintext "apple" with security key "snow" and using algorithm "Vignere cipher" is "*scdhw*".

1.10.1(B) Playfair Cipher

✍ ***Definition :*** *The Playfair cipher is a symmetric-key based encryption technique that uses digraph substitution cipher.*

This technique encrypts the pairs of alphabets (digraphs), instead of single alphabets as in the case of simple substitution ciphers like Caesar cipher. The Playfair cipher is thus significantly harder to break. It involves 625 combinations of alphabet pairs instead of just 26 in the case of single alphabets. Hence, the regular cryptanalysis techniques such as the frequency analysis are harder to perform.

Algorithm

1. Start by creating a 5 x 5 key square by choosing a key and filling rest of the places by the remaining alphabets such that any alphabet occurs only once in the 5 x 5 square. 5 x 5 will cover up only 25 alphabets. Hence, i and j are combined and treated as 1 position. This would then cover all the 26 alphabets in the 5 x 5 square.
2. Take the plaintext and remove any punctuations, special characters, numbers, etc. such that only alphabets remain in the plaintext. Then make pairs of the alphabets in the plaintext. If you have just one alphabet left out, use X to make a pair. Any pairs that have the same alphabets are also replaced by a X.
3. Use the pairs in plaintext to substitute with the key square positions. Locate the alphabets in the key square and follow the substitution rules :
   (a) If the alphabets appear on the same row of the key square, replace them with the alphabets to their immediate right respectively (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).
   (b) If the alphabets appear on the same column of the key square, replace them with the alphabets immediately below respectively (If the alphabet is at the bottom most position, wrap around to the take the topmost alphabet of the column).
   (c) If the alphabets are in different rows and columns, replace the pair with the alphabets on the same row respectively but at the corners of the rectangle defined by the original pair.

Ex. 1.10.1 : Use Playfair cipher to encrypt the word "greet" using the key "moon mission".

Soln. :

Step 1 : Construct the key square.

- Unique alphabets from the given key "moon mission" are → *m, o, n, i* and *s*. *i* and *j* are combined into one cell in the key square. Rest of the alphabets are filled serially such that the alphabets already in the key square (the key in the first row) are not repeated. This gives the following 5 x 5 key square.

m	o	n	i/j	s
a	b	c	d	e
f	g	h	k	l
p	q	r	t	u
v	w	x	y	z

Step 2: Arrange plaintext.

- The plaintext to encrypt is "greet". It is ordered as the following pairs.

  "gr", "ex", "et"

- The 2nd occurrence of *e* in *ee* is replaced with *x* to give "*ex*".

Step 3 : Apply substitution based on the key square and the plaintext pairs.

"*gr*" is encrypted as following.

m	o	n	i/j	s
a	b	c	d	e
f	g	h	k	l
p	q	r	t	u
v	w	x	y	z

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
- The alphabet in the corner of the rectangle of the same row as *g* is *h*.
- The alphabet in the corner of the rectangle of the same row as *r* is *q*.
- Hence, "*gr*" is encrypted as "*hq*".
- Now, pick the next plaintext pair "*ex*".

m	o	n	i/j	s
a	b	c	d	e
f	g	h	k	l
p	q	r	t	u
v	w	x	y	z

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
- The alphabet in the corner of the rectangle of the same row as *e* is *c*.
- The alphabet in the corner of the rectangle of the same row as *x* is *z*.
- Hence, "*ex*" is encrypted as "*cz*".
- Now, pick the next plaintext pair "*et*".

m	o	n	i/j	s
a	b	c	d	e
f	g	h	k	l
p	q	r	t	u
v	w	x	y	z

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
- The alphabet in the corner of the rectangle of the same row as *e* is *d*.
- The alphabet in the corner of the rectangle of the same row as *t* is *u*.
- Hence, "*et*" is encrypted as "*du*".
- Hence, the plaintext "*greet*" is encrypted as "*hqczdu*" using Playfair cipher using the key "*moon mission*".

Ex. 1.10.2 : Use Playfair cipher to encrypt the plaintext "Why, don't you?" using the key "keyword".

Soln. :

Step 1 : Construct the key square.

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

Step 2 : Arrange the plaintext into pairs.

(remove all punctuations).

"wh", "yd", "on", "ty", "ou"

Step 3 : Apply substitution for each plaintext pair.

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

"wh" is *"yi"*.

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

"yd" is *"ea"*.

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

"*on*" is "*es*".

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

"*ty*" is "*vk*".

k	e	y	w	o
r	d	a	b	c
f	g	h	i/j	l
m	n	p	q	s
t	u	v	x	z

"*ou*" is "*ez*".

Hence, "Why, don't you?" is encrypted as "yieaesvkez" using the key "keyword".

Ex. 1.10.3 : Encrypt "The key is hidden under the door" using Playfair cipher with keyword "domestic".

MU - Dec. 15, 5 Marks

Soln. :

Step 1 – Construct the 5 x 5 key square.

i and j are combined into one cell in the key square. Rest of the alphabets are filled serially such that the alphabets already in the key square (the key in the first row) are not repeated. This gives the following 5 x 5 key square.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

Step 2 – Arrange the plaintext into pairs.

(remove all punctuations).

"th", "ek", "ey", "is", "hi", "dd", "en", "un", "de", "rt", "he", "do", "or"

Step 3 – Apply substitution for each plaintext pair.

"th" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *t* is *c*.
- The alphabet in the corner of the rectangle of the same row as *h* is *f*.

Hence, "th" is encrypted as "cf".

"ek" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets appear on the same column of the key square. You need to replace them with the alphabets immediately below respectively (If the alphabet is at the bottom most position, wrap around to the take the topmost alphabet of the column).

- The alphabet below *e* is *a*
- The alphabet below *k* is *r*

Hence, "*ek*" is encrypted as "*ar*".

"ey" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets appear on the same column of the key square. You need to replace them with the alphabets immediately below respectively (If the alphabet is at the bottom most position, wrap around to the take the topmost alphabet of the column).

- The alphabet below *e* is *a*
- The alphabet below *y* is *e*

Hence, "*ey*" is encrypted as "*ae*".

"is" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *i* is *b*.
- The alphabet in the corner of the rectangle of the same row as *s* is *o*.

Hence, "*is*" is encrypted as "*bo*".

"hi" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *h* is *g*.
- The alphabet in the corner of the rectangle of the same row as *i* is *c*.

Hence, "*hi*" is encrypted as "gc".

"dd" is encrypted as "dx" as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *d* is *m*
- The alphabet in the corner of the rectangle of the same row as *x* is *v*

Hence, "*dd*" is encrypted as "*mv*".

"en" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *e* is *d*.
- The alphabet in the corner of the rectangle of the same row as *n* is *r*.

Hence, "*en*" is encrypted as "*dr*".

"un" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets appear on the same row of the key square. You need to replace them with the alphabets to their immediate right respectively (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).

- The alphabet to the right of *u* is *n*.
- The alphabet to the right of *n* is *p*.

Hence, "*un*" is encrypted as "np".

"de" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets appear on the same row of the key square. You need to replace them with the alphabets to their immediate right respectively (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).

- The alphabet to the right of *d* is *o*.
- The alphabet to the right of *e* is *s*

Hence, "*de*" is encrypted as "*os*".

"rt" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *r* is *n*
- The alphabet in the corner of the rectangle of the same row as *t* is *a*

Hence, "*rt*" is encrypted as "*na*".

"he" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

The alphabet in the corner of the rectangle of the same row as *h* is *k*

The alphabet in the corner of the rectangle of the same row as *e* is *m*

Hence, "*he*" is encrypted as "*km*".

"do" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets appear on the same row of the key square. You need to replace them with the alphabets to their immediate right respectively (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).

The alphabet to the right of *d* is *o*

The alphabet to the right of *o* is *m*

Hence, "*do*" is encrypted as "*om*".

"or" is encrypted as following.

d	o	m	e	s
t	i/j	c	a	b
f	g	h	k	l
n	p	q	r	u
v	w	x	y	z

The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.

- The alphabet in the corner of the rectangle of the same row as *o* is *e*

- The alphabet in the corner of the rectangle of the same row as *r* is *p*

Hence, "*or*" is encrypted as "*ep*".

Hence, the plaintext "The key is hidden under the door" using Playfair cipher with keyword "domestic" is encrypted as "cfarae bogc mv dr np osna km om ep".

1.10.1(C) Hill Cipher

✍ ***Definition :*** *The Hill cipher is a polygraphic substitution cipher based on linear algebra.*

By polygraphic, we mean that it can work on substitution for up to 3-alphabets at a time. It arranges the key and the plaintext into a matrix format and their multiplication undergoes the mod 26 operation to find the resultant ciphertext.

Algorithm

1. Arrange the key and the plaintext in a matrix format. Use the Table 1.10.4 for assigning numbers to alphabets for matrix operations. Note that you need to create the plaintext matrix according to the given key matrix such that multiplication is possible. The number of columns in the key matrix must be equal to the number of rows in the plaintext matrix. If the plaintext is larger, break it up into multiple matrices and apply further steps on each of the plaintext matrix using the key. To fill the matrix, in case you are short of matrix elements, you may use zeroes.

Table 1.10.4 : Alphabet Positions

Alphabet	Number	Alphabet	Number
A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16
E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

2. Carry out multiplication of the key and the plaintext matrix.
3. Perform mod 26 operation on the resultant multiplication.
4. Use the Table 1.10.4 again to convert numbers back to alphabets. These alphabets represent the ciphertext.

Ex. 1.10.4 : Encrypt the message "Exam" using the Hill cipher with the key $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$.

Soln. :

Plaintext "*Exam*" when converted into a number matrix would be $\begin{bmatrix} 4 & 0 \\ 23 & 12 \end{bmatrix}$.

Multiply the Key and Plaintext matrices.

$$\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix} \times \begin{bmatrix} 4 & 0 \\ 23 & 12 \end{bmatrix} = \begin{bmatrix} 128 & 48 \\ 181 & 84 \end{bmatrix}$$

Perform mod 26 operation on the result.

$$\begin{bmatrix} 128 & 48 \\ 181 & 84 \end{bmatrix} \times \text{mod } 26 = \begin{bmatrix} 24 & 22 \\ 25 & 6 \end{bmatrix}$$

Converting the numbers from mod operation back to alphabets you get "*YZWG*". Hence, encrypting the plaintext "*Exam*" using the given key gives ciphertext "*YZWG*".

Ex. 1.10.5 : Encrypt the message "DEF" using the Hill cipher with the key $\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 8 & 7 \end{bmatrix}$.

Soln. :

Plaintext "*DEF*" when converted into a number matrix would be

Multiply the Key and Plaintext matrices.

$$\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 8 & 7 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 47 \\ 40 \\ 76 \end{bmatrix}$$

Perform mod 26 operation on the result.

$$\begin{bmatrix} 47 \\ 40 \\ 76 \end{bmatrix} \times \text{mod } 26 = \begin{bmatrix} 21 \\ 14 \\ 24 \end{bmatrix}$$

Converting the numbers from mod operation back to alphabets you get "*VOY*". Hence, encrypting the plaintext "*DEF*" using the given key gives ciphertext "*VOY*".

1.10.1(D) Affine Cipher

✍ ***Definition :*** *Affine cipher is a type of monoalphabetic substitution cipher, wherein each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter.*

Each letter is encrypted using the Affine function *(Ax + B) mod 26*. Values (A, B) are called coefficients of Affine function. You use the position value of each alphabet and use the Affine function to calculate the position value of the corresponding encrypted letter. Then, you substitute each plaintext letter with the corresponding encrypted letter from the table.

Let's see an example.

Ex. 1.10.6 : Encrypt the plaintext message "SECURITY" using affine cipher with the key pair (3, 7). Decrypt to get back original plaintext. **MU - May 19, 10 Marks**

Soln. :

Here the coefficients of Affine function are (3, 7). So, A = 3 and B = 7.

Create a table of all alphabets using the formula *(Ax + B) mod 26*. Use A = 3 and B = 7 for this computation.

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B) mod 26	Encrypted Alphabet (Position from Affine calculation)
0	A	(3 * 0 + 7) mod 26 = 7	H
1	B	(3 * 1 + 7) mod 26 = 10	K
2	C	(3 * 2 + 7) mod 26 = 13	N
3	D	(3 * 3 + 7) mod 26 = 16	Q
4	E	(3 * 4 + 7) mod 26 = 19	T
5	F	(3 * 5 + 7) mod 26 = 22	W
6	G	(3 * 6 + 7) mod 26 = 25	Z
7	H	(3 * 7 + 7) mod 26 = 2	C
8	I	(3 * 8 + 7) mod 26 = 5	F
9	J	(3 * 9 + 7) mod 26 = 8	I
10	K	(3 * 10 + 7) mod 26 = 11	L
11	L	(3 * 11 + 7) mod 26 = 14	O
12	M	(3 * 12 + 7) mod 26 = 17	R
13	N	(3 * 13 + 7) mod 26 = 20	U
14	O	(3 * 14 + 7) mod 26 = 23	X
15	P	(3 * 15 + 7) mod 26 = 0	A
16	Q	(3 * 16 + 7) mod 26 = 3	D
17	R	(3 * 17 + 7) mod 26 = 6	G
18	S	(3 * 18 + 7) mod 26 = 9	J
19	T	(3 * 19 + 7) mod 26 = 12	M
20	U	(3 * 20 + 7) mod 26 = 15	P
21	V	(3 * 21 + 7) mod 26 = 18	S
22	W	(3 * 22 + 7) mod 26 = 21	V
23	X	(3 * 23 + 7) mod 26 = 24	Y
24	Y	(3 * 24 + 7) mod 26 = 1	B
25	Z	(3 * 25 + 7) mod 26 = 4	E

Once the table is ready, you can substitute each letter in the plaintext with its corresponding encrypted letter.

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B) mod 26	Encrypted Alphabet (Position from Affine calculation)
0	A	(3 * 0 + 7) mod 26 = 7	H
1	B	(3 * 1 + 7) mod 26 = 10	K
2	C	(3 * 2 + 7) mod 26 = 13	N
3	D	(3 * 3 + 7) mod 26 = 16	Q
4	E	(3 * 4 + 7) mod 26 = 19	T
5	F	(3 * 5 + 7) mod 26 = 22	W
6	G	(3 * 6 + 7) mod 26 = 25	Z
7	H	(3 * 7 + 7) mod 26 = 2	C
8	I	(3 * 8 + 7) mod 26 = 5	F
9	J	(3 * 9 + 7) mod 26 = 8	I
10	K	(3 * 10 + 7) mod 26 = 11	L
11	L	(3 * 11 + 7) mod 26 = 14	O
12	M	(3 * 12 + 7) mod 26 = 17	R
13	N	(3 * 13 + 7) mod 26 = 20	U
14	O	(3 * 14 + 7) mod 26 = 23	X
15	P	(3 * 15 + 7) mod 26 = 0	A
16	Q	(3 * 16 + 7) mod 26 = 3	D
17	R	(3 * 17 + 7) mod 26 = 6	G
18	S	(3 * 18 + 7) mod 26 = 9	J
19	T	(3 * 19 + 7) mod 26 = 12	M
20	U	(3 * 20 + 7) mod 26 = 15	P
21	V	(3 * 21 + 7) mod 26 = 18	S
22	W	(3 * 22 + 7) mod 26 = 21	V
23	X	(3 * 23 + 7) mod 26 = 24	Y
24	Y	(3 * 24 + 7) mod 26 = 1	B
25	Z	(3 * 25 + 7) mod 26 = 4	E

So,

S → J

E → T

C → N

U → P

R → G

I → F

T → M

Y → B

So, the plaintext SECURITY, when encrypted using Affine cipher using the Affine coefficients of (3, 7), give JTNPGFMB as the encrypted text.

Note here that if you have to decrypt the ciphertext using Affine cipher, you follow the same approach. You first create the table using the Affine coefficients and then substitute the encrypted letters with their corresponding plaintext letters.

Difference between Monoalphabetic and Polyalphabetic Ciphers

Note a key difference between monoalphabetic cipher and polyalphabetic cipher;

- For repeated characters, in monoalphabetic cipher, ciphertext is same (for example, plaintext y is ciphertext b as per example we chose earlier) whereas;
- For polyalphabetic cipher, repeated plaintext characters need not lead to the same ciphertext (for example, there are two instances of p in plaintext word apple in the polyalphabetic cipher example. One is encrypted as c and another is encrypted as d).

So, polyalphabetic ciphers are stronger than monoalphabetic since they usually give different ciphertext values for repeated characters in plaintext and hence are less prone to frequency analysis attack. Frequency analysis attack tries to find a correlation between plaintext and ciphertext and determine the security key. For example, the attacker might guess that y is encrypted as b in the monoalphabetic cipher, so it could mean the security key is "shift next by 3". Once the attacker determines the key, converting any ciphertext back to plaintext is a trivial (very simple) task.

For Cryptography to be successful, keeping the key secret is very important.

1.10.2 Transposition Techniques

Q. Explain with examples, keyed and keyless transposition ciphers.	MU - May 16, 5 Marks
Q. Discuss any one transposition cipher with example. List their merits and demerits.	MU - May 19, 4 Marks

In this operation, the position of characters is jumbled up (mixed up) like a letter arranging game. For example, the plaintext "apple" could be transposed into ciphertext as "elpap". Note that all the characters in plaintext are also present in the ciphertext but at different positions. For example, position of "a" in plaintext is 1 whereas in ciphertext it is 4. Note that it is a very simplistic example. Various complex mathematical transposition algorithms are usually used in cryptography. Transposition can be carried out using two techniques – Keyed and Keyless. Let us learn about them.

1.10.2(A) Keyed Transposition Cipher

✍ ***Definition :*** *In keyed transposition, a random key is used to describe the transposition sequence and carry out the transposition.*

This is also called Columnar Transposition Cipher.

Algorithm

1. Arrange the plaintext in a column under the given key.
2. Rearrange the plaintext column-wise in key's alphabetic order.

Ex. 1.10.7 : Use the key "ENCRYPT" to encrypt the plaintext "Save the king from attack" using transposition cipher.

Soln. : Draw a table and arrange the key and the plaintext under the key column-wise. Mark the alphabets in the key in their order alphabetically. For example, for the given key, the alphabet "*C*" comes first in the order of 26 alphabets. Then comes "*E*" and hence marked 2. Likewise mark all the alphabets in the key according to their occurrence in the alphabets.

E	N	C	R	Y	P	T
2	3	1	5	7	4	6
s	a	v	e	t	h	e
k	i	n	g	f	r	o
m	a	t	t	a	c	k

Read the columns in order.

- Take column C marked as 1st in order : vnt
- Take column E marked as 2nd in order : skm
- Take column N marked as 3rd in order : aia

Follow likewise to get the ciphertext as "*vntskmaiahrcegteoktfa*".

Ex. 1.10.8 : Use the key "SORROW" to encrypt the plaintext "Demonetization tonight" using transposition cipher.

Soln. :

Draw a table and arrange the key and the plaintext as following. Mark the order of the columns from left to right in case of repeated key characters. Pad the columns with "x" to fill the columns if the plaintext does not fill the table completely.

S	O	R	R	O	W
5	1	3	4	2	6
d	e	m	o	n	e
t	i	z	a	t	i
o	n	t	o	n	i
g	h	t	*x*	*x*	*x*

The resulting ciphertext is "*eimhntnxmzttoaoxdtogeiix*".

1.10.2(B) Keyless Transposition Cipher

✍ ***Definition :*** *In keyless transposition, a transposition sequence is described without a random key.*

One such example of Keyless Transposition Cipher is Railfence Cipher. Railfence cipher uses the rail size as a key and does not use a random key as such. It can be easily broken.

Algorithm

1. Based on the rail size, arrange the plaintext.
2. Rearrange the plaintext row-wise to get the ciphertext.

Ex. 1.10.9 : Encrypt the plaintext "Save the king from attack" using Railfence cipher. Assume a suitable rail size.

Soln. :

Assuming a rail size of 3. All it means is that it would have 3 rows. Arrange the plaintext in rails one alphabet at a time.

rail 1 →	s				t				i				r				t				k
rail 2 →		a		e		h		k		n		f		o		a		t		c	
rail 3 →			v				e				g				m				a		

Rearrange the plaintext rail-wise (row-wise) to get the ciphertext. Start from rail 1, then rail 2 and finally rail 3. Here, the ciphertext would be "*stirtkaehknfoatcvegma*".

Ex. 10.7.10 : Encrypt the plaintext "Demonetization tonight" using Railfence cipher. Assume the rail size of 4.

Soln. : Arrange the plaintext in 4 rails (rows).

rail 1 →	d						t						o						g		
rail 2 →		e				e		i				i		n				i		h	
rail 3 →			m		n				z		t				t		n				t
rail 4 →				o						a						o					

The resulting ciphertext is "*dtogeeiinihmnzttntoao*".

1.11 Methods of Encryption

As you know, encryption is primarily driven by two components – Keys and Algorithms. Based on the number of keys used in the encryption and decryption process, the encryption methods can be classified as,

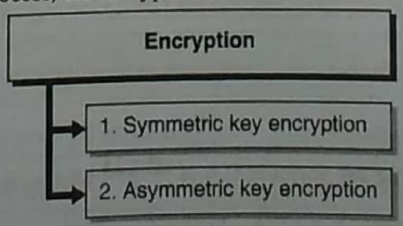

Fig. 1.11.1 : Encryption

1.11.1 Symmetric Key Encryption

Symmetric means same.

✍ ***Definition :*** *In Symmetric Key Encryption, the key used for encryption is same as the key used for decryption.*

The keys are identical. The sender as well as the receiver must have exactly the same key to encrypt or decrypt. As an example, symmetric key is like your regular lock key. The same key can be used for locking the door as well as unlocking the door.

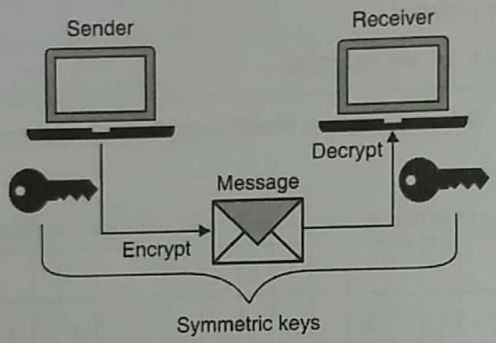

Fig. 1.11.2 : Symmetric Key Encryption

As you understand, a symmetric key is unique between a sender and a receiver. If there are more entities involved and each require to secretly communicate with the another, you end up having multiple keys. Let's take an example, suppose there are 4 friends – A, B, C and D and each one of them require communicating with one another secretly. It is obvious that you cannot share the keys between a pair of friends with another pair of friends. So, if A and B share a key, B and C cannot share the same key because C would also know the secret key between A and B and can then decrypt communication between A and B. So, you would require several keys to ensure that each pair of sender and receiver have a unique key. So, how many keys would you need?

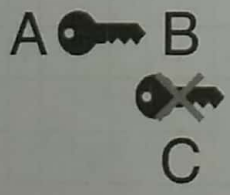

Fig. 1.11.3 : Symmetric keys between a pair should not be shared

You would need below keys (one for each pair of sender and receiver) :

1. A <> B
2. A <> C
3. A <> D
4. B <> C
5. B <> D
6. C <> D

This can be effectively calculated using the formula $K = N \frac{(N-1)}{2}$ where, N is the number of entities that need to secretly communicate. So, in the above example, $K = 4 * (4 - 1) / 2 = 6$.

If we have 100 entities, it would require 100 * (100 – 1) /2 = 4,950 keys! What if the entire world wants to communicate with each other? Can you imagine? I will come back to it later on and answer that for you.

Another problem here is how does A send the key she is using to B? If the sender and receiver have to use the same key, there should be a way to securely transfer the key. Isn't it? For example, if you have to give your house keys to your friend, you probably exchange hands in person. You don't leave the key somewhere that could potentially be picked / looked by someone else other than your friend. I will answer this question as well later on.

Some of the examples of symmetric key algorithm are DES, AES and Blowfish.

Advantages of Symmetric Keys

1. Computationally faster than the asymmetric keys.
2. Hard to break if the key used is long.

Disadvantages of Symmetric Keys

1. Requires a secure mechanism to exchange keys.
2. Each pair of sender and receiver require a unique key.
3. Provides only confidentiality but not authenticity and non-repudiation.

1.11.2 Asymmetric Key Encryption

Unlike symmetric keys,

✍ ***Definition :*** *In Asymmetric Key Encryption, there are two keys that are mathematically related. If one is used for encryption, then only the corresponding other key can be used for decryption.*

Asymmetric means not equal. In the asymmetric system, two mathematically related keys work as key pair. If you use one key for encryption, then you need the other key in the pair for decryption. The encrypting key cannot be used for decrypting.

Note : Asymmetric Keys based Cryptography is also called as Public Key Cryptography.

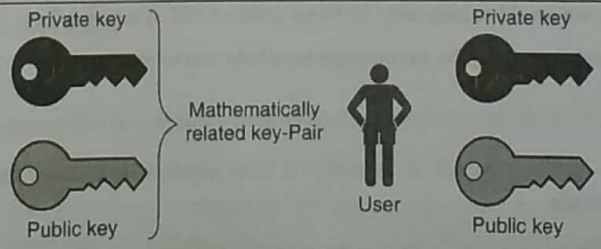

Fig. 1.11.4 : Asymmetric Key Encryption

Let's call one of the keys as the Public Key and its counterpart in the pair as the Private Key. A user owns both the keys. The public key is known to the world while the private key is known only to the user.

Key Used	Corresponding Key Required	Security Service Provided
Encryption – Public Key	Decryption – Private Key	Confidentiality
Private Key	Public Key	Authentication and Non-repudiation

Let us understand these two use cases of the asymmetric keys.

Use Case 1 - User A wants to send a secret message to User B

- User A knows : User A's Public Key, User A's Private Key, User B's Public Key.
- User B knows : User A's Public Key, User B's Public Key, User B's Private Key.

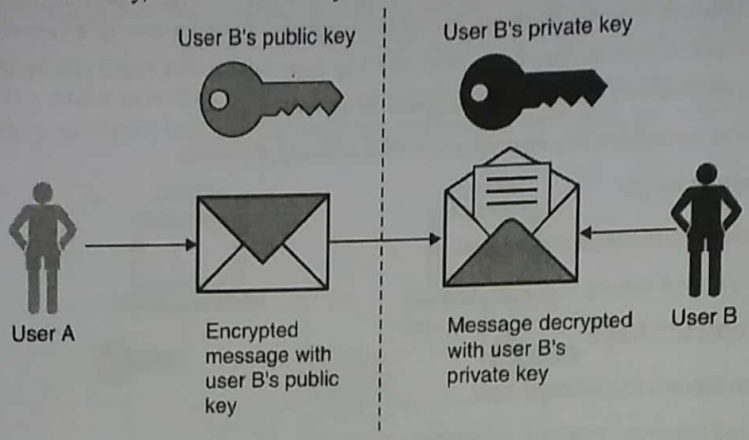

Fig. 1.11.5 : Encryption and Decryption using Public Key Cryptography

- User A wants to send an encrypted message such that only User B can read it. User A encrypts the message with User B's Public Key. Now, because User A used User B's Public Key, she is sure that only User B can decrypt the message as decryption would require the corresponding Private Key and only User B knows about her Private Key.
- Hence, in this scenario you find that asymmetric keys as well can be used to send encrypted messages as you saw in the case of symmetric keys. One core difference to note here is that User A need not know User B's Private Key to send her an encrypted message. A separate key distribution problem does not exist as Public Keys are known to the world and only the user needs to know and protect her Private Keys.
- You also see that you require only two keys per entity for encrypted communication. So, for 100 people to send each other encrypted messages, we would require only 200 keys unlike 4,095 symmetric keys (recall from our discussion on symmetric keys in the previous section)! So, asymmetric keys help you to overcome two of the limitations of symmetric keys :
  1. Key distribution
  2. Number of keys to manage
- Hence, I answer the question for you that I asked in the previous section – how do we manage keys if the whole world wants to communicate with each other? The answer is using asymmetric keys!

Use Case 2 - Proving authenticity and non-repudiation

- The second use case of asymmetric keys is for proving authenticity of an entity. This is highly used today for server validation. Have you seen "https" in front of website address? That is one of the examples of this use case.
- Suppose, you want to do an online transaction. How do you ensure that the bank's website address you are interacting with is the right one? That's precisely what asymmetric keys help you solve as well.

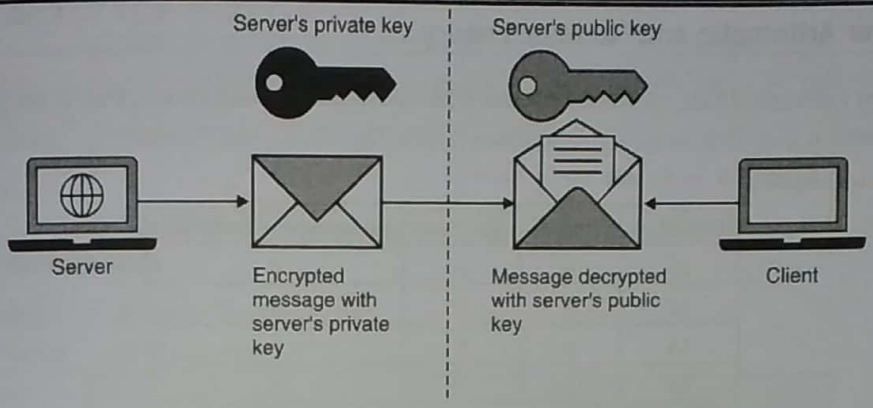

Fig. 1.11.6 : Non-repudiation using Public Key Cryptography

- Client wants to authenticate the server before it begins the transaction. It sends the server a plaintext message and asks it to encrypt it with its Private Key. The server encrypts the message that the client sent with its Private Key and sends it back to the client.
- Client uses the world-known Public Key of the server and decrypts the message it received from the server. If the message gets successfully decrypted and it matches with what the client sent earlier to the server to encrypt, the client has now validated that it is indeed interacting with the authentic server because except the authentic server, no one else would have known server's Private Key. The client is satisfied, and it begins the secure transaction after having established the server's authenticity.
- Hence, you find that asymmetric keys could effectively be used for authentication and non-repudiation. Some examples of algorithms that use asymmetric keys are RSA, ECC, Diffie-Hellman, and El Gamal.

Advantages of Asymmetric Keys

1. Easy key distribution.
2. Less number of keys to manage.
3. Can also be used for providing authentication and non-repudiation.

Disadvantages of Asymmetric Keys

1. Slower than symmetric keys.
2. Requires significant CPU power due to complex mathematical relation between the keys.

1.11.3 Comparison between Symmetric and Asymmetric Keys

Comparison Attribute	Symmetric Keys	Asymmetric Keys
Speed	High	Low
Complexity	Low	High
Number of keys	High	Low
Key Distribution	Problematic	Easier
Security Services	Confidentiality	Confidentiality, Authenticity, Non-repudiation

1.12 Modular Arithmetic and Number Theory

Definition : *The modular arithmetic deals with operations on integers specifically around remainders from division.*

Let's take a few examples.

Dividend	Divisor	Quotient	Remainder (Modulus)
15	5	3	0
15	4	3	3
15	3	5	0
15	2	7	1
15	1	15	0

So, for example, number 15 when divided by 2, gives you Quotient of 7 and Remainder of 1. This can be mathematically written as 15 mod 2.

Note : Before you proceed, try finding out a few mods (remainders from division) for numbers of your choice.

1. Congruence Property

- Two numbers are said to be in congruence modulo, if they give out the same mod.

  For example,

  $15 \bmod 2 = 1$

  $17 \bmod 2 = 1$

- Hence, 15 is congruent to 17 *modulo* 2 i.e. 15 and 17 when undergo mod operation with 2, they both give same result of 1. They can be mathematically denoted as,

  $15 \equiv 17 \pmod 2$

2. Modular Addition

$(A + B) \bmod C = (A \bmod C + B \bmod C) \bmod C$

Let, A = 12, B = 15 and C = 5

Left Hand Side	Right Hand Side
$= (12 + 15) \bmod 5 = 27 \bmod 5 = 2$	$= (12 \bmod 5 + 15 \bmod 5) \bmod 5$ $= (2 + 0) \bmod 5 = 2 \bmod 5$ $= 2$

3. Modular operation on Negative numbers

- If you come across modular operation on a negative number, make the number positive by repetitively adding mod until it becomes positive.
- **For example :** If you have to find −13 mod 5, keeping adding 5 to −13 until you get a positive number. So,

  $-13 + 5 = -8 + 5 = -3 + 5 = 2$.

  Now, do mod on the positive number you got. In this case, −13 mod 5 becomes 2 mod 5. Hence, the answer is 2.

4. Modular Subtraction

$(A - B) \bmod C = (A \bmod C - B \bmod C) \bmod C$

Let, A = 12, B = 15 and C = 5

Left Hand Side	Right Hand Side
= (12 – 15) mod 5 = –3 mod 5 = 2	= (12 mod 5 – 15 mod 5) mod 5 = (2 – 0) mod 5 = 2 mod 5 = 2

5. Modular Multiplication

(A * B) mod C = (A mod C * B mod C) mod C

Let, A = 12, B = 15 and C = 5

Left Hand Side	Right Hand Side
= (12 * 15) mod 5 = 180 mod 5 = 0	= (12 mod 5 * 15 mod 5) mod 5 = (2 * 0) mod 5 = 0 mod 5 = 0

6. Modular Inverse

- Modular arithmetic does not have division operation. However, it has inverse. Inverse of a general number is 1 divided by that number.
- **For example :** Inverse of 2 is ½. In other words, a number when multiplied by its inverse would give 1. So, 2 multiplied by its inverse ½ would give 2 * ½ = 1.
- So, modular inverse of A mod C is the value of B such that when A is multiplied by B and the mod C operation is carried out, it gives 1. Mathematically, it can be written as,

$$A * B \equiv 1 \pmod{C}$$

Note : To understand the above, refer to the congruency equation. mod C operation on A*B should be same as mod C operation on 1.

Let's take an example.

Suppose, you have to find modular inverse of 12 mod 5. Here, A = 12 and C = 5. Let's assume value of B from 0 onwards until we find (A*B) mod C = 1.

Value of B	Operation	Result
0	(12 * 0) mod 5	0
1	(12 * 1) mod 5	2
2	(12 * 2) mod 5	4
3	**(12 * 3) mod 5**	**1**

Hence, modular inverse of 12 mod 5 is 3.

7. Prime Numbers

- These are whole numbers which are greater than 1 and are only divisible by 1 and itself.
- **For example :** 2, 3, 5, 7, 11 and various others.

8. Coprime Numbers

- Two integers a and b are said to be relatively prime, mutually prime, or coprime (also written co-prime) if the only positive integer (factor) that divides both of them is 1.
- **For example :** 5 and 7 are coprime because their common factor is only 1 whereas 14 and 18 are not coprime because their common factor can also be 2.

9. Discrete Logarithm

- If a is an arbitrary integer relatively prime to n and g is a primitive root of n, then there exists exactly one number μ such that $a = g^{\mu} \pmod{n}$.
- The number μ is then called the discrete logarithm of a with respect to the base g modulo n and is denoted

$$\mu = \text{ind}_g a \pmod{n}.$$

- Discrete logarithms are quickly computable in a few special cases. However, no efficient method is known for computing them in general. Several important algorithms in public-key cryptography use discrete logarithms.

1.13 Greatest Common Divisor (GCD)

Greatest Common Divisor (GCD) of two positive integers is the largest integer that can fully divide both the integers. For example, GCD (5, 10) is 5. GCD of (11, 13) is 1. GCD is usually found out by finding the factors of the respective integers and then choosing the common highest factor.

For example, to find GCD (24, 70)

- Factors of 24 = 2 * 2 * 2 * 3 → Factors could be 2, 3, 4, 6, 8, 12, 24
- Factors of 70 = 2 * 5 * 7 → Factors are only 2, 5, 7

Hence, the largest common factor is 2. Hence, GCD (24, 70) = 2.

1.13.1 Euclid's or Euclidean Algorithm

Finding GCD for smaller numbers is quite straight forward. But, when it comes to finding GCD of large numbers, it might be a complex task. This is precisely where Euclid's (or Euclidean) algorithm helps.

Euclid's algorithm states that,

$$\gcd(a,b) = \gcd(a \bmod b, b) \quad \text{if } a > b$$

$$\gcd(a,b) = \gcd(a, b \bmod a) \quad \text{if } b > a$$

Ex. 1.13.1 : Find gcd(50, 65) using Euclidean algorithm.

Soln. :

$\gcd(50,65) = \gcd(50,65 \bmod 50)$ [because 65 is greater than 50]

$= \gcd(50,15)$

$= \gcd(50 \bmod 15,15)$ [because 50 is greater than 15]

$= \gcd(5,15)$

$= \gcd(5,15 \bmod 5)$ [because 15 is greater than 5]

$= \gcd(5,0)$ [stop here once the mod of a term becomes 0]

$= 5$ [is the gcd(50,65)]

Ex. 1.13.2 : Find gcd(464, 238) using Euclidean algorithm.

Soln. :

$$\begin{aligned}\gcd(464,238) &= \gcd(464 \bmod 238, 238)\\ &= \gcd(226,238)\\ &= \gcd(226,238 \bmod 226)\\ &= \gcd(226,12)\\ &= \gcd(226 \bmod 12,12)\\ &= \gcd(10,12)\\ &= \gcd(10,12 \bmod 10)\\ &= \gcd(10,2)\\ &= \gcd(10 \bmod 2,2)\\ &= \gcd(0,2)\\ &= 2\end{aligned}$$

Ex. 1.13.3 : Find gcd(105, 80).

Soln. :

$$\begin{aligned}\gcd(105,80) &= \gcd(105 \bmod 80,80)\\ &= \gcd(25,80 \bmod 25)\\ &= \gcd(25 \bmod 5,5)\\ &= \gcd(0,5)\\ &= 5\end{aligned}$$

1.13.2 Extended Euclidean Algorithm

The Extended Euclidean Algorithm can be used to find the gcd of two numbers, and also to simultaneously express the gcd as a linear combination of these numbers. It helps to find values of coefficients x and y such that to satisfy the following equation :

$$ax + by = \gcd(a,b)$$

In the extended algorithm, the computation involves several entities. First, let's define them:

I = index : This would just be used to iterate (repeat) the process.

Note : If b is greater than a, then assume a = b and b = a. Swap the values to avoid negatives.

- **r = remainder :** Temporary placeholder variable for a and b.

  r would be calculated as $r_{i+1} = r_{i-1} - q_i r_i$

- q would be calculated as $q_{i+1} = r_{i-1} / r_i$
- s = Temporary placeholder for values while deriving coefficient x.

  s would be calculated as $s_{i+1} = s_{i-1} - q_i s_i$

- t = Temporary place holder for values while deriving coefficient y.

  t would be calculated as $t_{i+1} = t_{i-1} - q_i t_i$

- x would be $s_{i+1} = \frac{b}{\gcd(a, b)}$

  y would be $t_{i+1} = \frac{a}{\gcd(a, b)}$

Ex. 1.13.4 : For a = 161 and b = 42, calculate gcd(a, b) and also the values of x and y to satisfy the extended Euclidean algorithm.

Soln.:

$r_0 = 161$ and $r_1 = 42$

i starts with 0

First draw the initial table.

Index i	quotient q for i	Remainder r for i	S for i	t for i
0		161	1	0
1		42	0	1

Start steps.

Index i	quotient q for i	Remainder r for i	S for i	t for i
0		161	1	0
1		42	0	1
2	161 / 42 = 3	161 – 3 * 42 = 35	1 – 3 * 0 = 1	0 – 3 * 1 = –3

For the highlighted row,

- i = 1 (we are just doing the calculation for 2nd row, hence the value of i is still 1)
- q_1 can we written as q_i where i = 1
  - So, $q_1 = r_{i-1} / r_i = r_0 / r_1 = 161 / 42 = 3$
- r_2 can be written as r_{i+1} where i = 1
  - So, $r_2 = r_{i-1} - q_i r_i$ which means $r_2 = r_0 - q_1 * r_1$
  - $r_2 = 161 - 3 * 42$

  $r_{i-1} = r_0$ which is 161

  $r_i = r_1$ which is 42
- Similarly, s_2 can be written as s_{i+1} where i = 1
  - So, $s_2 = s_{i-1} - q_i s_i = s_0 - q_1 s_1 = 1 - 3*0 = 1$
- Similarly, t_2 can be written as t_{i+1} where i = 1
  - So, $t_2 = t_{i-1} - q_i t_i = t_0 - q_1 t_1 = 0 - 3*1 = -3$
- Similarly proceed to the next step.

Index i	quotient q for i	Remainder r for i	S for i	t for i
0		161	1	0
1		42	0	1
2	161 / 42 = 3	161 – 3 * 42 = 35	1 – 3 * 0 = 1	0 – 3 * 1 = –3
3	42 / 35 = 1	42 – 1 * 35 = 7	0 – 1 * 1 = –1	1 – 1 * –3 = 4

Here again :

$q_i = q_2$ = where i = 2

- So, $q_2 = r_{i-1} / r_i = r_1 / r_2 = 42 / 35 = 1$

$r_3 = r_1 - q_2 * r_2 = 42 - 1 * 35 = 7$

$s_3 = s_1 - q_2 * s_2 = 0 - 1 * 1 = -1$

$t_3 = t_1 - q_2 * t_2 = 1 - 1 * -3 = 4$

Index i	quotient q for i	Remainder r for i	S for i	t for i
0		161	1	0
1		42	0	1
2	161 / 42 = 3	161 - 3 * 42 = 35	1 - 3 * 0 = 1	0 - 3 * 1 = -3
3	42 / 35 = 1	42 - 1 * 35 = 7	0 - 1 * 1 = -1	1 - 1 * -3 = 4
4	35 / 7 = 5	35 - 5 * 7 = 0	Do not calculate	Do not calculate

$q_i = q_3$ where i = 3

- So, $q_3 = r_2 / r_3 = 35 / 7 = 5$

$r_4 = r_2 - q_3 * r_3 = 35 - 5 * 7 = 0$

- Do not calculate s and t once you get r = 0
- Last calculated s and t become x and y respectively
- Last calculated r becomes gcd

So, according to extended Euclidean algorithm, for numbers 161 and 42

gcd(161, 42) = 7

In the equation ax + by = gcd(161,42)

x = -1

y = 4

You can verify the answer by putting the values in the equation.

Left-hand side:

ax + by = 161*-1 - 42*4

= -161 + 168

= 7 [which is equal to gcd(161,42)]

Ex. 1.13.5 : For a = 256 and b = 5004, calculate gcd(a, b) and also the values of x and y to satisfy the extended Euclidean algorithm.

Soln. :

First note that b > a. Hence, lets swap the values to make the calculations simple. We would re-swap them in the final answer.

So, assume a = 5004 and b = 256

Index i	quotient q for i	Remainder r for i	S for i	t for i
0		5004	1	0
1		256	0	1

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		5004	1	0
1		256	0	1
2	5004 / 256 = 19	5004 – 19 * 256 = 140	1 – 19*0 = 1	0 – 19*1 = –19
3	256 / 140 = 1	256 – 1 * 140 = 116	0 – 1*1 = –1	1 – 1*–19 = 20
4	140 / 116 = 1	140 – 1 * 116 = 24	1 – 1*1 = 2	–19 – 1*20 = –39
5	116 / 24 = 4	116 – 24 * 4 = 20	–1 – 4*2 = –9	20 – 39*4 = 176
6	24 / 20 = 1	24 – 20 * 1 = **4**	2 – 9*1 = **11**	–39 – 1*176 = **–215**
7	20 / 4 = 5	20 – 4* 5 = 0	Do not calculate	Do not calculate

gcd(256, 5004) = 4

We originally swapped the value. So, re-swap it.

Hence, x = –215 and y = 11

Putting it in the equation, you get,

Left-Hand Side :

$$ax + by = 256 * -215 + 5004 * 11$$
$$= -55{,}040 + 55{,}044$$
$$= 4 \text{ [which is equal to right hand side = gcd(256,5004)]}$$

Ex. 1.13.6 : For a = 86 and b = 14, calculate gcd(a, b) and also the values of x and y to satisfy the extended Euclidean algorithm.

Soln. :

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		86	1	0
1		14	0	1
2	$\frac{86}{14} = 6$	86 – 6*14 = **2**	1 – 6*0 = **1**	0 – 6*1 = **–6**
3	$\frac{14}{2} = 7$	14 – 2*7 = 0	Do not calculate	Do not calculate

gcd(86,14) = 2

x = 1, y = –6

To verify, let's put the above values in the equation:

$$ax + by = 86*1 - 14*6$$
$$= 86 - 84$$
$$= 2 \text{ [this is the gcd value that we got for 86,14]}$$

Ex. 1.13.7 : For a = 999 and b = 9, calculate gcd(a, b) and also the values of x and y to satisfy the extended Euclidean algorithm.

Soln. :

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		999	1	0
1		**9**	**0**	**1**
2	999 / 9 = 111	999 – 111*9 = 0	Do not calculate	Do not calculate

gcd(999,9) = 9

x = 0, y = 1

Let's put those values in the equation:

ax + by= 999 * 0 + 1*9

= 9 [which matches the gcd value we got for 999,9]

✍ ***Exam Tip :*** *It would perhaps be easy for you to calculate the first two columns q and r until you get 0 in r. That way you are focusing on one set of calculation at a time. Once you have q and r computed in the first two columns, calculate s and t by just substitution the values of q and r in the respective equations.*

1.13.3 Multiplicative Inverse using Extended Euclidean Algorithm

If you recall our discussion from the previous section on modular inverse, you understand what inverse operation is. One application of the extended Euclidean Algorithm is to find out multiplicative inverse.

✍ ***Definition :*** *A modular multiplicative inverse of an integer a is an integer x such that the product ax is congruent to 1 with respect to the modulus m.*

In modular arithmetic, it can be written as $ax \equiv 1 \pmod m$

According to the extended Euclidean Algorithm,

$ax + my = \gcd(a,m) = 1$

$ax + my = 1$

$ax - 1 = (-y)m$

Dividing both sides by (mod m)

$ax \pmod m - 1 \pmod m = (-y)m \pmod m$

$ax \pmod m - 1 \pmod m = 0$

$ax \equiv 1 \pmod m$

Note : The multiplicative inverse of a modulo m exists if and only if a and m are coprime (i.e., if gcd(a, m) = 1)

So, if you come across a question where it is asked to calculate multiplicative inverse such that gcd(a, m) is not 1, do not attempt to solve the problem. Just calculate the gcd and show that the numbers are not coprime and hence the multiplicative inverse does not exist.

Ex. 1.13.8 : Find multiplicate inverse of 24140 mod 40902.

Soln. :

gcd(24140, 40902) = gcd(24140,40902 mod 24140)
= gcd(24140,16762)
= gcd(24140 mod 16762,16762)
= gcd(7378,16762)
= gcd(7378,16762 mod 7378)
= gcd(7378,2006)
= gcd(7378 mod 2006,2006)
= gcd(1360,2006)
= gcd(1360,2006 mod 1360)
= gcd(1360 mod 646,646)
= gcd(68,646 mod 68)
= gcd(68 mod 34,34)
= gcd(0,34)
= 34

Here you find that gcd(24140, 40902) = 34. Hence, multiplicative inverse of 24140 mod 40902 does NOT exist.

Note : You can also calculate gcd using tabular method as you learnt in the extended Euclidean algorithm section to avoid repeating the gcd steps to calculate x and y if gcd = 1 does exist.

Ex. 1.13.9 : Find multiplicate inverse of 8 mod 11.

Soln. :

Since 8 < 11, let's swap the values for simplicity.

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		11	1	0
1		8	0	1

Calculate next steps.

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		11	1	0
1		8	0	1
2	11 / 8 = 1	11 – 8*1 = 3	1 – 1*0 = 1	0 – 1*1 = –1
3	8 / 3 = 2	8 – 3*2 = 2	0 – 2*1 = –2	1 – 2*–1 = 3
4	3 / 2 = 1	3 – 2*1 = **1**	1 – 1*–2 = **3**	–1 – 1*3 = –4
5	2 / 1 = 2	2 – 1*2 = 0	Do not calculate	Do not calculate

Let's re-swap the values.

Hence, x = – 4 and y = 3

Putting the values in the extended Euclidean algorithm, you get

ax + by = 1

$8(-4) + 11(3) = 1$

Since, you have to find multiplicative inverse in mod 11, divide both sides by mod 11.

$8(-4) \bmod 11 + 11(3) \bmod 11 = 1 \bmod 11$

$8(-4) \bmod 11 + 0 = 1$

Recall our discussion on calculating mod for negative numbers. You need to keep adding mod until the number turns positive and then calculate mod on the positive number you got.

$-4 + 11 = 7$

$7 \bmod 11 = 7$

Hence, $8(7) \bmod 11 = 1$

So, multiplicative inverse of 8 mod 11 is 7.

Note : You can also test your solution. If there are mistakes, re-visit the steps you took. In this example, 8*7 = 56 and 56 mod 11 = 1. Hence, you find that 7 is indeed multiplicative inverse of 7 in mod 11.

Ex. 1.13.10 : Find multiplicate inverse of 1234 mod 4321.

Soln. :

Since 1234 < 4321, let's swap the values for simplicity.

Index i	quotient q for i	Remainder r for i	s for i	t for i
0		4321	1	0
1		1234	0	1
2	4321 / 1234 = 3	4321 – 3*1234 = 619	1 – 3*0 = 1	0 – 3*1 = –3
3	1234 / 619 = 1	1234 – 1*619 = 615	0 – 1*1 = –1	1 – 1*–3 = 4
4	619 / 615 = 1	619 – 1*615 = 4	1 – 1*–1 = 2	–3 – 1*4 = –7
5	615 / 4 = 153	615 – 4*153 = 3	–1 – 153*2 = –307	4 – 153*–7 = 1075
6	4 / 3 = 1	4 – 1*3 = **1**	2 – 1*–307 = **309**	–7 – 1*1075 = **–1082**
7	3 / 1 = 3	3 – 3*1 = 0	Do not calculate	Do not calculate

Let's re-swap the values.

Hence, $x = -1082$ and $y = 309$

Putting it in the equation,

$ax + by = 1$

$1234(-1082) + 4321(309) = 1$

Dividing both sides by mod 4321, you get

$1234(-1082) \bmod 4321 + 4321(309) \bmod 4321 = 1 \bmod 4321$

$1234(-1082) \bmod 4321 + 0 = 1$

Convert –1082 to positive

$-1082 + 4321 = 3239$

$3239 \bmod 4321 = 3239$

Hence,

$1234(3239) \bmod 4321 = 1$

Or 3239 is multiplicative modular inverse of 1234 in mod 4321.

Review Questions

Here are a few review questions to help you gauge your understanding of this chapter. Try to attempt these questions and ensure that you can recall the points mentioned in the chapter.

[A] Computer Security Concepts

Q. 1 Explain the terms Assets, Controls, Threats, Vulnerabilities, Risk and Exposure with examples. **(8 Marks)**

Q. 2 Draw a block diagram depicting the relation between Assets, Controls, Threats, Vulnerabilities, Risk and Exposure. **(6 Marks)**

Q. 3 With many security controls, you can reduce the risk to zero. Comment. **(4 Marks)**

Q. 4 Describe the three pillars of security with examples. **(8 Marks)**

Q. 5 Describe the three goals of security with examples. **(8 Marks)**

Q. 6 Write a short note on confidentiality. **(4 Marks)**

Q. 7 Write a short note on integrity. **(4 Marks)**

Q. 8 Write a short note on availability. **(4 Marks)**

Q. 9 Explain the terms Identification, Authentication, Authorisation, Accountability and Non-repudiation with example. **(8 Marks)**

Q. 10 Draw a block diagram depicting the relation between Identification, Authentication, Authorisation and Accountability. **(6 Marks)**

[B] OSI Model

Q. 11 Draw a chart for various Security Services provided at various OSI layers. **(6 Marks)**

Q. 12 What is OSI Model? List a few Security Services and Mechanisms for each layer. **(8 Marks)**

Q. 13 Write a short note on Network Security Model. **(6 Marks)**

[C] Types of Security Attacks

Q. 14 Classify security attacks and briefly explain each attack. **(8 Marks)**

Q. 15 Describe Replay Attack with a block diagram. **(8 Marks)**

Q. 16 Describe Denial of Service (DoS) Attack with a block diagram. **(8 Marks)**

Q. 17 Describe Fabrication Attack with a block diagram. **(4 Marks)**

Q. 18 Write a short note on traffic analysis. **(6 Marks)**

Q. 19 Write a short note on eavesdropping. **(6 Marks)**

Q. 20 Compare Active and Passive attacks. **(8 Marks)**

[D] Classical Encryption Techniques

Q. 21 With an example, explain how you can use substitution for encryption. **(6 Marks)**

Q. 22 With an example, explain how you can use transposition for encryption. **(6 Marks)**

Q. 23	Take an example of your choice and work it through explaining Vignere Cipher.	**(8 Marks)**
Q. 24	Take an example of your choice and work it through explaining Playfair Cipher.	**(8 Marks)**
Q. 25	Take an example of your choice and work it through explaining Hill Cipher.	**(8 Marks)**
Q. 26	Take an example of your choice and work it through explaining Keyed Transposition Cipher.	**(8 Marks)**
Q. 27	Take an example of your choice and work it through explaining Keyless Transposition Cipher.	**(8 Marks)**
Q. 28	Write a short note explaining the difference between monoalphabetic and polyalphabetic cipher.	**(6 Marks)**
Q. 29	Write a short note on steganography.	**(4 Marks)**
Q. 30	Compare cryptography with steganography.	**(4 Marks)**

[F] Methods of Encryption

Q. 31	Describe Symmetric Key Encryption.	**(6 Marks)**
Q. 32	Write advantages and disadvantages of Symmetric Key Encryption.	**(4 Marks)**
Q. 33	Describe Asymmetric Key Encryption.	**(6 Marks)**
Q. 34	Write advantages and disadvantages of Asymmetric Key Encryption.	**(4 Marks)**
Q. 35	Compare symmetric and asymmetric encryption.	**(6 Marks)**

❑❑❑