**CS6701 CRYPTOGRAPHY AND NETWORK SECURITY L T P C 3 0 0 3**

**UNIT I INTRODUCTION & NUMBER THEORY 10**

Services, Mechanisms and attacks-the OSI security architecture- Network security model-Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography). FINITE FIELDS AND NUMBER THEORY: Groups, Rings, Fields-Modular arithmetic- Euclid‟s algorithm-Finite fields-Polynomial Arithmetic –Prime numbers-Fermat‟s and Euler‟s theorem- Testing for primality -The Chinese remainder theorem- Discrete logarithms.

**UNIT IIBLOCK CIPHERS & PUBLIC KEY CRYPTOGRAPHY 10**

Data Encryption Standard-Block cipher principles-block c pher modes of operation-Advanced Encryption Standard (AES)-Triple DES-Blowfish-RC5 algorithm. **Public key cryptography:** Principles of public key cryptosystems-The RSA algorithm-Key management - Diffie Hellman Key exchange- Elliptic curve arithmetic-Elliptic curve cryptography.

**UNIT IIIHASH FUNCTIONS AND DIGITAL SIGNATURES 8**

Authentication requirement – Authentication function – MAC – Hash function – Security of hash function and MAC –MD5 - SHA - HMAC – CMAC - Digital signature and authentication protocols – DSS – I Gamal – Schnorr.

**UNIT IVSECURITY PRACTICE & SYSTEM SECURITY 8**

Authentication applications – Ke beros – X.509 Authentication services - Internet Firewalls for Trusted System: Roles of Firewalls – Firewall related terminology- Types of Firewalls - Firewall d signs – SET for E-Commerce Transactions. Intruder – Intrusion detection system – Virus and related threats – Countermeasures – Firewalls design principles – Trusted systems – Practical implementation of cryptography and security.

**UNIT VE-MAIL, IP & WEB SECURITY 9**

**E-mail Security:** Security Services for E-mail-attacks possible through E-mail - establishing keys privacy-authentication of the source-Message Integrity-Non-repudiation-Pretty Good Privacy-S/MIME. **IPSecurity:** Overview of IPSec - IP and IPv6-Authentication Header-Encapsulation Security Payload (ESP)-Internet Key Exchange (Phases of IKE, ISAKMP/IKE Encoding). **Web Security:** SSL/TLS Basic Protocol-computing the keys- client authentication-PKI as deployed by SSLAttacks fixed in v3-Exportability-Encoding-Secure Electronic Transaction (SET).

**TOTAL: 45 PERIODS**

**Text Book**


1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013. (UNIT I, II, III, IV).

2. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002. (UNIT V).


**References**


1. Behrouz A. Ferouzan, “Cryptography & Network Secur ty”, Tata McGraw Hill, 2007.

2. Man Young Rhee, “Internet Security: Cryptographic Principles”, “Algorithms and Protocols”, Wiley Publications, 2003.

3. Charles Pfleeger, “Security in Computing”, 4th Edition, Prentice Hall of India, 2006.

4. Ulysess Black, “Internet Security Protocols”, Pearson Education Asia, 2000.

5. Charlie Kaufman and Radia Perlman, Mike Speciner, “Network Security, Second Edition, Private Communication in Public World”, PHI 2002.

6. Bruce Schneier and Neils Ferguson, “Practical Cryptography”, First Edition, Wiley Dreamtech India Pvt Ltd, 2003.

7. Douglas R Simson “Cryptography – Theory and practice”, First Edition, CRC Press, 1995.

8. http://nptel.ac in/

## UNIT I INTRODUCTION & NUMBER THEORY 10

Services, Mechanisms and attacks-the OSI security architecture-Network security model-Classical Encryption techniques (Symmetric cipher model, substitution techniques, transposition techniques, steganography).FINITE FIELDS AND NUMBER THEORY: Groups, Rings, Fields-Modular arithmetic-Euclid‟s algorithm-Finite fields- Polynomial Arithmetic –Prime numbers-Fermat‟s and Euler‟s theorem-Testing for primality -The Chinese remainder theorem- Discrete logarithms.

COMPUTER SECURITY CONCEPTS

Computer Security

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information / data, and telecommunications)

**Confidentiality**

- Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized
- Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

**Integrity**

- Data integrity: Assures that information and programs are changed only in a specified and authorized manner.
- System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

**Availability**

Assures that systems work promptly and service is not denied to authorized users.

**CIA Triad**

Confidentiality

Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

Integrity

Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.

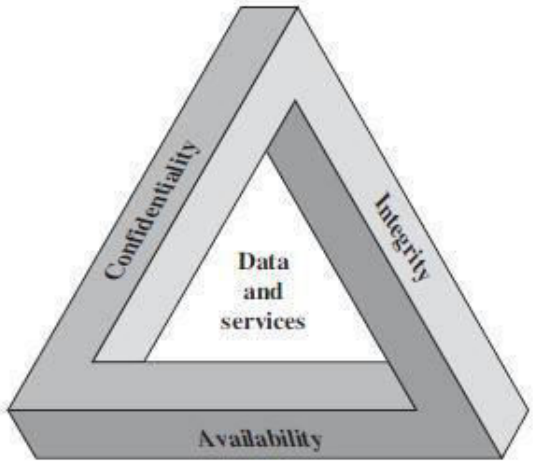


Figure 1.1 The Security Requirements Triad

Availability
- Ensuring timely and reliable access to and use of information
- A loss of availability is the disruption of access to or use of information or an information system.

Authenticity
- The property of being genuine and being able to be verified and trusted

Accountability
- The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity

## TYPES OF ATTACKS

### OSI Security Architecture

### Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit vulnerability.

- **Attack**

  An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.

- The OSI security architecture is useful to managers as a way of organizing the task of providing security. This architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms.

The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security mechanism:** A process that is designed to detect, prevent, or recover from a security attack.
- **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of o e or more security mechanisms to provide the service.

- **Security Attacks:**A useful means of classifying security attacks, used both in X.800 and RFC 4949, is in terms of *passive attacks* and *active attacks.* A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

**Passive Attacks**

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that isbeing transmitted. Two types of passive attacks are the release of message contents and traffic analysis

The **release of message contents:** A telephone conversation, an elect onic mail message, and a transferred file may cont in sensitive or confidential information. An opponent should be prevented from learning the contents of these transmissions.

**Traffic analysis:** Suppose that there is a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption.If encryption protection in place, an opponent might still be able to observe the pattern of these messages. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

**Active Attacks**

- Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.
- A **masquerade** takes place when one entity pretends to be a different entity.
- **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

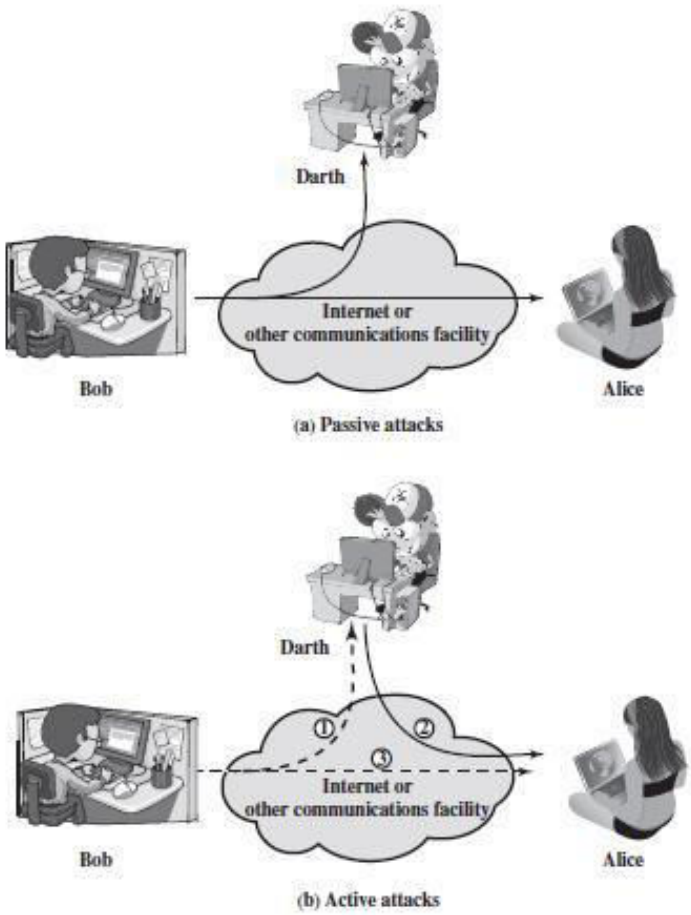


**Figure: Active and Passive attacks**

- **Modification of messages** simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.
- The **denial of service** prevents or inhibits the normal use or management of communications facilities.

Passive attacks are difficult to detect; measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely

**Security Services**

- X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. X.800 divides these services into five categories and fourteen specific services.

**1. Authentication**

The authentication service is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. Two specific authentication services are defined in X.800:

**Peer entity authentication:** Provides for the corroboration of the identity of a peer entity in an association. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.

**Data origin authentication:** Provides for the corroboration of the source of data unit.

It does not provide protection against the duplication or modification of data units.

**2. Access Control**

In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

**3. Data Confidentiality**

Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time.

| **AUTHENTICATION** | **DATA INTEGRITY** |
|---|---|
| The assurance that the communicating entity is theone that it claims to be.<br><br>**Peer Entity Authentication**<br>Used in association with a logical connection toprovide confidence in the identity of the entitiesconnected.<br><br>**Data-Origin Authentication**<br>In a connectionless transfer, provides assurance that the source of received data is as claimed.<br><br>**ACCESS CONTROL**<br>The prevention of unauthorized use of a resource.<br><br>**DATA CONFIDENTIALITY**<br>The protection of data from unauthorized disclosure.<br><br>**Connection Confidentiality**<br>The protection of all user data on a connection.<br><br>**Connectionless Confidentiality**<br>The protection of all user data in a single data block | The assurance that data received are exactly assent by an authorized entity.<br><br>**Connection Integrity with Recovery**<br>Provides for the ntegrity of all user data on aconnection and detects any modification, insertion,deletion, or replay of a y data within an entire datasequence, with recovery attempted.<br><br>**Connection Integrity without Recovery**<br>As above, but provides only detection without recovery.<br><br>**Selective-Field Connection Integrity**<br>Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.<br><br>**Connectionless Integrity**<br>Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited |

| | |
|---|---|
| **AUTHENTICATION**<br>The assurance that the communicating entity is theone that it claims to be.<br><br>**Peer Entity Authentication**<br>Used in association with a logical connection toprovide confidence in the identity of the entitiesconnected.<br><br>**Data-Origin Authentication**<br>In a connectionless transfer, provides assurance that the source of received data is as claimed.<br><br>**ACCESS CONTROL**<br>The prevention of unauthorized use of a resource.<br><br>**DATA CONFIDENTIALITY**<br>The protection of data from unauthorized disclosure.<br><br>**Connection Confidentiality**<br>The protection of all user data on a connection.<br><br>**Connectionless Confidentiality**<br>The protection of all user data in a single data block<br>**Selective-Field Confidentiality**<br>The confidentiality of selected fields within the user data on a connection or in a single data block. | form of replay detection may be provided.<br><br>**Selective-Field Connectionless Integrity**<br>Provides for the integrity of selected fields within a single connectionless data block.<br><br>**NONREPUDIATION**<br>Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.<br>**Nonrepudiation, Origin**<br>Proof that the message was sent by the specified party.<br>**Nonrepudiation, Destination**<br>Proof that the message was received by the specified party. |

**Table:Security Services (X.800)**

## 4. Data Integrity

As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays. The connection-oriented integrity service addresses both message stream modification and denial of service.

On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.

**5. Nonrepudiation**

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

## Security Mechanisms

The mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service.

**Table:Security Mechanisms (X.800)**

| SPECIFIC SECURITY MECHANISMS | PERVASIVE SECURITY MECHANISMS |
|---|---|
| May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services. | Mechanisms that are not specific to any particular OSI security service or protocol layer. |
| **Encipherment**<br>The use of mathematical algorithms to transform data into a form that is not readily intelligible. | **Trusted Functionality**<br>That which is perceived to be correct with respect tosome criteria. |
| **Digital Signature**<br>Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unitto prove the source and integrity of the data unit and protect against forgery. | **Security Label**<br>The marking bound to a resource that names or designates the security attributes of that resource. |
| **Access Control**<br>A variety of mechanisms that enforce access rights to resources. | **Event Detection**<br>Detection of security-relevant events. |
| **Data Integrity** | **Security Audit Trail**<br>Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities. |
| | **Security Recovery** |

| | |
|---|---|
| A variety of mechanisms used to assure the integrity of a data unit or stream of data units.<br><br>**Authentication Exchange**<br><br>A mechanism intended to ensure the identity of an entity by means of information exchange.<br><br>**Traffic Padding**<br><br>The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.<br><br>**Routing Control**<br><br>Enables selection of particular Physically secure routes for certain Data and allows routing changes, especially when a breach of security is suspected.<br><br>**Notarization**<br>The use of a trusted third party to assure certain properties of adata exchange. | Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions. |

X.800 distinguishes between reversible encipherment mechanisms and irreversible encipherment mechanisms. A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible encipherment mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

# CLASSICAL ENCRYPTION TECHNIQUES- SUBSTITUTION AND TRANSPOSITION

## Substitution Techniques

A substitution technique is o e in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

## Caesar Cipher

The earliest known, and the simplest, use of a substitution cipher was by Julius Caes r. The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.

- For example, plain: meet me after the toga party
  cipher: PHHW PH DIWHU WKH WRJD SDUWB
- The alphabet is wrapped around, so that the letter following Z is A.
- The transformation are defined by listing all possibilities, as follows:
  plain: 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
  cipher: 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

Let us assign a numerical equivalent to each letter:

| a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

A numerical equivalent to each letter:

- Then the algorithm can be expressed as follows. For each plaintext letter $p$, substitute the ciphertext letter $C$:
  $$C = E(3, p) = (p + 3) \bmod 26$$
- A shift may be of any amount, so that the general Caesar algorithm is $C = E(k, p) = (p + k) \bmod 26$
  where $k$ takes on a value in the ange 1 to 25. The decryption algorithm is simply
  $$p = D(k, C) = (C - k) \bmod 26$$
- If it is known that a given c phertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys. The following figureshows the results of applying this strategy to the example ciphertext.
- Three important characteristics of this problem enabled us to use a bruteforce crypta alysis:
  1. The encryption and decryption algorithms are known.
  2. There are only 25 keys to try.

3. The language of the plaintext is known and easily recognizable.

| KEY | PHHW PH DIWHU WKH WRJD SDUWB |
|---|---|
| 1 | oggv og chvgt vjg vqic rctva |
| 2 | nffu nf bgufs uif uphb qbsuz |
| 3 | meet me after the toga party |
| 4 | ldds ld zesdq sgd snfz ozqax |
| 5 | kccr kc ydrcp rfc rmey nyprw |
| 6 | jbbq jb xcqbo qeb qldx mxoqv |
| 7 | iaap ia wbpan pda pkcw lwnpu |
| 8 | hzzo hz vaozm ocz ojbv kvmot |
| 9 | gyyn gy uznyl nby niau julns |
| 10 | fxxm fx tymxk max mhzt itkmr |
| 11 | ewwl ew sxlwj lzw lgys hsjlq |
| 12 | dvvk dv rwkvi kyv kfxr grikp |
| 13 | cuuj cu qvjuh jxu jewq fqhjo |
| 14 | btti bt puitg iwt idvp epgin |
| 15 | assh as othsf hvs hcuo dofhm |
| 16 | zrrg zr nsgre gur gbtn cnegl |
| 17 | yqqf yq mrfqd ftq fasm bmdfk |
| 18 | xppe xp lqepc esp ezrl alcej |
| 19 | wood wo kpdob dro dyqk zkbdi |
| 20 | vnnc vn jocna cqn cxpj yjach |
| 21 | ummb um inbmz bpm bwoi xizbg |
| 22 | tlla tl hmaly aol avnh whyaf |
| 23 | skkz sk glzkx znk zumg vgxze |
| 24 | rjjy rj fkyjw ymj ytlf ufwyd |
| 25 | qiix qi ejxiv xli xske tevxc |

**Figure: Brute force cryptanalysis of caesar cipher**

**Monoalphabetic Ciphers**

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. Before proceeding, the term *permutation can be defined*.

- A permutation of a finite set of elements *S* is an ordered sequence of all the elements of *S*, with each element appearing exactly once.
  For example, if *S* = {a, b, c}, there are six permutations of *S*:

  abc, acb, bac, bca, cab, cba

- In general, there are *n*! permutations of a set of *n* elements, because the first element can be chosen in one of *n* ways, the second in *n* - 1 ways, the third in *n* – 2 ways, and so on.
  plain: 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 Caesar

  cipher: 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

- If, instead, the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 * 1026 possible keys.
- This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a mono alphabetic substitution cipher,

because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

- Mono alphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.
- A countermeasure is to provide multiple substitutes known as homophones, for a single letter. For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone assigned to a letter in rotation or randomly.

**Playfair Cipher**

The best-known multiple-letter encryption cipher is the Playfair, which treats diagrams in the plaintext as si gle units and translates these units into ciphertext diagrams.

The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers‟s *Have His Carcase.*

| | | | | |
|---|---|---|---|---|
| M | O | N | A | R |
| C | H | Y | B | D |
| E | F | G | I/J | K |
| L | P | Q | S | T |
| U | V | W | X | Z |

- In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix ith the remaining letters in alphabetic order. The letters I and J count as one letter.

- Plaintext is encrypted two letters at a time, according to the following rules:
  1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.
  2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.
  3. Two plaintext letters that fall in the same column are each replaced by the letter beneath,with the top element of the column circularly following the last. For example, mu is encrypted as CM.
  4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM.

- The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 * 26 = 676 diagrams, so that identification of individual diagrams is more difficult. Furthermore, the relative frequencies of individual letters

exhibit a much greater range than that of diagrams, making frequency analysis much more difficult.

- For these reasons, the Playfair cipher was for a long time considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

**Hill Cipher**

Another interesting multi letter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929.

**The Hill Algorithm**

- This encryption algorithm takes *m* successive plaintext letters and substitutes for them *m* cipher text letters. The substitution is determined by *m* linear equations in which each character is assigned a numerical value (a = 0, b = 1,….., z = 25). For *m* = 3, the system can be described as

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3) \bmod 26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3) \bmod 26$$

$$c_3 = (k_{13}p_1 + k_{23}p_2 + k_{33}p_3) \bmod 26$$

This can be expressed in terms of row vectors and matrices:

$$\mathbf{C} = \mathbf{PK} \bmod 26$$

where**C** and **P** are row vectors of length 3 representing the plaintext and cipher text, and **K** is a 3 x 3 matrix representing the encryption key. Operations are performed mod 26.This is demonstrated as

- It is easily seen that if the matrix $\mathbf{K}^{-1}$ is applied to the cipher text, then the plaintext is recovered.

In general terms, the Hill system can be expressed as

$$\mathbf{C} = E(\mathbf{K}, \mathbf{P}) = \mathbf{PK} \bmod 26$$

$$\mathbf{P} = D(\mathbf{K}, \mathbf{C}) = \mathbf{CK}^{-1} \bmod 26 = \mathbf{PKK}^{-1} = \mathbf{P}$$

- As with Playfair, the strength of the Hill cipher is that it completely hides single-letter frequencies. Indeed, with Hill, the use of a larger matrix hides more frequency information. Thus, a 3 * 3 Hill cipher hides not only single-letter but also two-letter frequency information.
- Although the Hill cipher is strong against a ciphertext-only attack, it is easily broken with a known plaintext attack.

**Polyalphabetic Ciphers**

Another way to improve on the simple mono alphabetic technique is to use different mono alphabetic substitutions as one proceeds through the plaintext m ssage.

- The general name for this approach is poly alphabetic substitution cipher. All these techniques have the following features in common:
  1. A set of related mono alphabetic substitution rules is used.
  2. A key determines which particular rule is chosen for a given transformation.
- ***Vigenère Cipher***The best known, and one of the simplest, polyalphabetic ciphers is the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value 3.
- The Vigenère cipher can be expressed in the following manner. Assume a sequence of plaintext letters $P = p_0, p_1, p_2, \ldots, p_{n-1}$ and a key consisting of the sequence of letters $K = k_0, k_1, k_2, \ldots, k_{m-1}$, where typically $m < n$. The sequence of ciphertext letters $C = C_0, C_1, C_2, \ldots, C_{n-1}$ is calculated as follows:

  $C = C_0, C_1, C_2, \ldots, C_{n-1} = E(K, P) = E[(k_0, k_1, k_2, \ldots, k_{m-1}), (p_0, p_1, p_2, \ldots, p_{n-1})]$

  $= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \ldots, (p_{m-1} + k_{m-1}) \bmod 26,$

  $(p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \ldots, (p_{2m-1} + k_{m-1}) \bmod 26, \ldots$

- Thus, the first letter of the key is added to the first letter of the plaintext, mod 26, the second letters are added, a d so on through the first *m* letters of the plaintext. For the next *m* letters of the plaintext, the key letters are repeated. This process continues until all of the plaintext sequence is encrypted. A general equation of the encryption process is

$$C_i = (p_i + k_i \bmod m) \bmod 26$$

A general equation for decrypt on is

$$p_i = (C_i - k_i \bmod m) \bmod 26$$

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword is *deceptive*, the message "we are discovered save yourself" is encrypted as

Key :*deceptivedeceptivedeceptive*

plaintext : wearediscoveredsaveyourself

ciphertext : ZICVTWQNGRZGVTWAVZHCQYGLMGJ

The strength of this cipher is that there are multiple cipher text letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost.

***Vernam Cipher*** The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it. Such a system was introduced by an AT&T engineer named Gilbert Vernam in 1918.

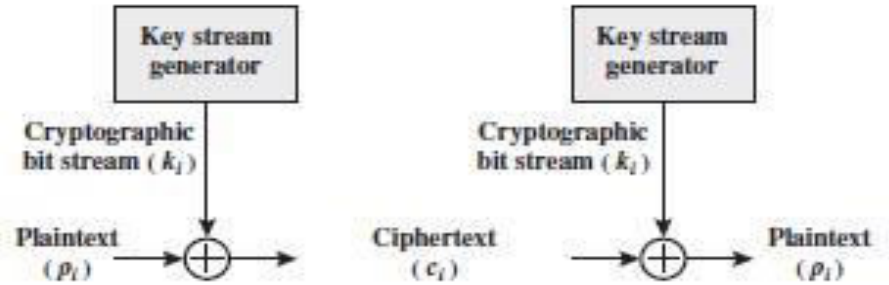


**Figure: Vernam Cipher**

- The system can be expressed as:

$$ci = pi \oplus ki$$

where $p_i$ = $i$th binary digit of plaintext

$k_i$ = $i$th binary digit of k y

$c_i$ = $i$th binary digit of ciphertext

$\oplus$ = exclusive-or (XOR) operation

Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation:

$$p_i = c_i \oplus k_i$$

**One-Time Pad**

- An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement to the Vernam cipher that yields the ultimate in security. Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated. In addition, the key is to be used to encrypt and decrypt a single message, and then is discarded.
- Each new message requires a new key of the same length as the new message. Such a scheme, known as a one-time pad, is unbreakable. It produces random output that bears no statistical relationship to the plaintext.
- Example:Consider a Vigenère scheme with 27 characters in which the twenty-seventh character is the space character, but with a one-time key that is as long as the message. Consider the ciphertext
  ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

Two different decryptions are shown using two different keys:

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: *pxlmvmsydofuyrvzwctnlebnecvgdupahfzzlmnyih*

plaintext: mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: *pftgpmiydgaxgoufhklllmhsqdqogtewbqfgyovuhwt*

plaintext: miss scarlet with the knife in the library

- The security of the one-time pad is entirely due to the randomness of the key. If the stream of characters that constitute the key is truly random, then the stream of characters that constitute the cipher text will be truly random. Thus, there are no patterns or regularities that a cryptanalyst can use to attack the ciphertext.
- The one-time pad offers complete security but, in practice, has two fundamental difficulties:
    1. There is the practical problem of making large quantities of random keys. Supplying truly random characters in this volume is a significant task.
    2. Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver. Thus, a mammoth key distribution problem exists.

- Because of these difficulties, the one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security The one-time pad is the only cryptosystem that exhibits what is referred to as *perfect secrecy*.

**Transposition Techniques**

- A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.
- The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.
- For example, to encipher the message “meet me after the toga party” with a rail fence of depth 2, we write the following:

m e m a t r h t g p r y
e t e f e t e o a a t

The encrypted message is

MEMATRHTGPRYETEFETEOAAT

- This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example,

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

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easilyreconstructed.

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

- To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is

  01 02 03 04 05 06 07 08 09 10 11 12 13 14

  15 16 17 18 19 20 21 22 23 24 25 26 27 28

- After the first transposition,

  03 10 17 24 04 11 18 25 02 09 16 23 01 08

  15 22 05 12 19 26 06 13 20 27 07 14 21 28

  which has a somewhat regular structure. But after the second transposition,

  17 09 05 27 24 16 12 07 10 02 22 20 03 25

  15 13 04 23 19 14 11 01 26 21 18 08 06 28

  This is a much less structured permutation and is much more difficult to cryptanalyze.

# FERMAT'S AND EULER'S THEOREM

Two theorems tht play important roles in public-key cryptography are Fermat"s theorem and Euler"s theorem.

## Fermat's Theorem

Fermat"s theorem states the following: If $p$ is prime and $a$ is a positive integer not divisible by $p$, then

$$a^{p-1} \equiv 1 \pmod{p}$$

***Proof:*** Consider the set of positive integers less than $p$: {1, 2, c, $p$ - 1}

*Proof:* Consider the set of positive integers less than $p$: $\{1, 2, \ldots, p-1\}$ and multiply each element by $a$, modulo $p$, to get the set $X = \{a \bmod p, 2a \bmod p, \ldots, (p-1)a \bmod p\}$. None of the elements of $X$ is equal to zero because $p$ does not divide $a$. Furthermore, no two of the integers in $X$ are equal. To see this, assume that $ja \equiv ka \pmod{p}$, where $1 \leq j < k \leq p-1$. Because $a$ is relatively prime[5] to $p$, we can eliminate $a$ from both sides of the equation [see Equation (4.3)] resulting in $j \equiv k \pmod{p}$. This last equality is impossible, because $j$ and $k$ are both positive integers less than $p$. Therefore, we know that the $(p-1)$ elements of $X$ are all positive integers with no two elements equal. We can conclude the $X$ consists of the set of integers $\{1, 2, \ldots, p-1\}$ in some order. Multiplying the numbers in both sets ($p$ and $X$) and taking the result mod $p$ yields

$$a \times 2a \times \cdots \times (p-1)a \equiv [(1 \times 2 \times \cdots \times (p-1)] \pmod{p}$$
$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

We can cancel the $(p-1)!$ term because it is relatively prime to $p$ [see Equation (4.5)]. This yields Equation (8.2), which completes the proof.

$a = 7, p = 19$
$7^2 = 49 \equiv 11 \pmod{19}$
$7^4 \equiv 121 \equiv 7 \pmod{19}$
$7^8 \equiv 49 \equiv 11 \pmod{19}$
$7^{16} \equiv 121 \equiv 7 \pmod{19}$
$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \pmod{19}$

≡

- This last equality is impossible, because $j$ and $k$ are both positive integers less than $p$. Therefore, ($p$ - 1) elements of $X$ are all positive integers with no two elements equal.
- We can conclude the $X$ consists of the set of integers {1, 2, …..,$p$ - 1} in some order. Multiplying the numbers in both sets ($p$ and $X$) and taking the result mod $p$ yields

$$a \times 2a \times \ldots\ldots\ldots \times (p-1)a \equiv [(1 \times 2 \times \ldots\ldots\ldots \times (p-1)] \pmod{p}$$

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

- We can cancel the ($p$ - 1)! term because it is relatively prime to $p$. This yields Equation, which completes the proof.

$a = 7, p = 19$

$7^2 = 49 \equiv 11 \pmod{19}$

$7^4 \equiv 121 \equiv 7 \pmod{19}$

$7^8 \equiv 49 \equiv 11 \pmod{19}$

$7^{16} \equiv 121 \equiv 7 \pmod{19}$

$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \pmod{19}$

- An alternative form of Fermat's theorem is also useful: If $p$ is prime and $a$ is a positive integer, then

$$a^p \equiv a \pmod{p}$$

**Euler's Totient Function**

Euler's totient function, f($n$), is defined as the number of positive integers less than $n$ and relatively prime to $n$. By convention, $\Phi(1) = 1$.

Determine $\Phi(37)$ and $\Phi(35)$.

Because 37 is prime, all of the positive integers from 1 through 36 are relatively

prime to 37. Thus $\Phi(37) = 36$.

To determine $\Phi(35)$, we list all of the positive integers less than 35 that are relatively

prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18

19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

There are 24 numbers on the list, so $\Phi(35) = 24$.

The below table lists the first 30 values of f($n$). The value f(1) is without meaning but is defined to have the value 1.

It should be clear that, for a prime number $p$,

$$\Phi(p) = p - 1$$

Now suppose that we have two prime numbers $p$ and $q$ with $p \neq q$. Then we can show that, for $n = pq$,

$$\Phi(n) = \Phi(pq) = \Phi(p) \times \Phi(q) = (p - 1) \times (q - 1)$$

To see that $\Phi$ (n) = $\Phi$ (p) × Φ (q), consider that the set of positive integers less that n is the set {1,……, (pq- 1)}. The integers in this set that are not relatively prime to n are the set {p, 2p,….., (q - 1)p} and the set {q, 2q,……, (p - 1)q}. Accordingly,

$$\Phi(n) = (pq- 1) - [(q - 1) + (p - 1)]$$

$$= pq- (p + q) + 1$$

$$= (p - 1) \times (q - 1)$$

**Euler's Theorem**

Euler‟s theorem states that for every *a* and *n* that are relatively prime:

$$a^{\Phi(n)} \equiv 1 (\text{mod } n)$$

***Proof:*** The above equation is true if *n* is prime, because in that case, $\Phi$ (*n*) = (*n*-1) and Fermat‟s theorem holds. However, it also holds for any integer *n*. $\Phi$ (*n*) is the number of positive integers less than *n* that are relatively prime to *n*.

Consider the set of such integers, labeled as

$$R = \{x_1, x_2, \ldots\ldots, x_{\Phi(n)}\}$$

- That is, each element $x_i$ of *R* is a unique positive integer less than *n* with gcd($x_i$, *n*) = 1.

Now multiply each element by *a*, modulo *n*:

$$S = \{(ax_1 \text{ mod } n), (ax_2 \text{ mod } n), \ldots\ldots\ldots, (_{()\text{mod } n)}\}$$

- The set *S* is a permutation of *R*, by the following line of reasoning:

  Because *a* is relatively prime to *n* and $x_i$ is relatively prime to *n*, $ax_i$ must also be relatively prime to *n*. Thus, all the members of *S* are integers that are less than *n* and that are relatively prime to *n*.

If $ax_i \text{mod } n = ax_j \text{mod } n$, then $x_i = x_j$.

## CHINESE REMINDER THEOREM

One of the most useful results of number theory is the **Chinese remainder theorem**(CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.

The CRT can be stated in several ways. Let M = k i=1 mi where the mi are pairwise relatively prime; that is, gcd(mi , mj ) = 1 for 1 … i, j … k, and i ≠ j. We can represent any integer A in ZM by a k-tuple whose elements are in Zmi using the following correspondence:

$$A \leftrightarrow (a_1, a_2, \ldots, a_k) \tag{8.7}$$

where $A \in Z_M$, $a_i \in Z_{m_i}$, and $a_i = A \bmod m_i$ for $1 \le i \le k$. The CRT makes two assertions.

1. The mapping of Equation (8.7) is a one-to-one correspondence (called a **bijection**) between $Z_M$ and the Cartesian product $Z_{m_1} \times Z_{m_2} \times \ldots \times Z_{m_k}$. That is, for every integer $A$ such that $0 \le A \le M$, there is a unique $k$-tuple $(a_1, a_2, \ldots, a_k)$ with $0 \le a_i < m_i$ that represents it, and for every such $k$-tuple $(a_1, a_2, \ldots, a_k)$, there is a unique integer $A$ in $Z_M$.
2. Operations performed on the elements of $Z_M$ can be equivalently performed on the corresponding $k$-tuples by performing the operation independently in each coordinate position in the appropriate system.

Let us demonstrate the **first assertion**. The transformation from $A$ to $(a_1, a_2, \ldots, a_k)$, is obviously unique; that is, each $a_i$ is uniquely calculated as $a_i = A \bmod m_i$. Computing $A$ from $(a_1, a_2, \ldots, a_k)$ can be done as follows. Let $M_i = M/m_i$ for $1 \le i \le k$. Note that $M_i = m_1 \times m_2 \times \ldots \times m_{i-1} \times m_{i+1} \times \ldots \times m_k$, so that $M_i \equiv 0 \pmod{m_j}$ for all $j \ne i$. Then let

$$c_i = M_i \times \left(M_i^{-1} \bmod m_i\right) \qquad \text{for } 1 \le i \le k \tag{8.8}$$

By the definition of $M_i$, it is relatively prime to $m_i$ and therefore has a unique multiplicative inverse mod $m_i$. So Equation (8.8) is well defined and produces a unique value $c_i$. We can now compute

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i\right) \pmod{M} \tag{8.9}$$

To show that the value of $A$ produced by Equation (8.9) is correct, we must

- The **second assertion** of the CRT, concerning arithmetic operations, follows from the rules for modular arithmetic. That is, the second assertion can be stated as follows: If

$$A \leftrightarrow (a_1, a_2, \ldots, a_k)$$
$$B \leftrightarrow (b_1, b_2, \ldots, b_k)$$

Then

$$(A + B) mod\ M \leftrightarrow \left((a_1 + b_1) mod\ m_1, \ldots, (a_k + b_k) mod\ m_k\right)$$
$$(A - B) mod\ M \leftrightarrow \left((a_1 - b_1) mod\ m_1, \ldots, (a_k - b_k) mod\ m_k\right)$$
$$(A \times B) mod\ M \leftrightarrow \left((a_1 \times b_1) mod\ m_1, \ldots, (a_k \times b_k) mod\ m_k\right)$$

One of the useful features of the Chinese remainder theorem is that it provides a way to manipulate (potentially very large) numbers mod M in terms of tuples of smaller numbers. This can be useful when M is 150 digits or more.

## The Euclidean Algorithm

One of the basic techniques of number theory is the Euclidean algorithm, which is a simple procedure for determining the greatest common divisor of two positive integers. Two integers are **relatively prime** if their only common positive integer factor is 1.

## Greatest Common Divisor

Nonzero $b$ is defined to be a divisor of $a$if $a = mb$ for some $m$, where $a$, $b$, and $m$ are integers. The notation gcd($a$, $b$) will be used to mean the **greatest common divisor** of $a$ and $b$. The greatest common divisor of $a$ and $b$ is the largest integer that divide both $a$ and $b$, definegcd(0, 0) = 0.

More formally, the positive integer $c$ is said to be the greatest common divisor of $a$ and $b$ if

1. $c$is a divisor of $a$ and of $b$.
2. Any divisor of $a$ and$b$ is a divisor of $c$.

An equivalent definition is the following:

$$\gcd(a, b) = \max[k, \text{ such that } k \mid a \text{ and } k \mid b]$$

The greatest common divisor should be positive,

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

In general, $\gcd(a, b) = \gcd(|a|, |b|)$.

Eg.gcd(60,24)=gcd(60,-24)=12

Also, because all nonzero integers divide 0, $\gcd(a, 0) = |a|$.

Two integers $a$ and$b$ are relatively prime if their only common positive integer factor is 1. This is equivalent to saying that $a$ and $b$ are relatively prime if $\gcd(a, b) = 1$.

## Finding the Greatest Common Divisor

An algorithm can be credited to Euclid for easily finding the greatest common divisor of two integers. Suppose we have integers $a$, $b$ such that $d = \gcd(a, b)$. Because $\gcd(|a|, |b|) = \gcd(a, b)$, there is no harm in assuming $a \geq b > 0$. Now dividing $a$ by $b$ and applying the division algorithm, we can

state: $a = q_1 b + r_1 \quad 0 \leq r_1 < b$

If it happens that $r_1 = 0$, then $b|a$and $d = \gcd(a, b) = b$. But if $r_1 \neq 0$, we can state that $d|r_1$. This is due to the basic properties of divisibility: the relations $d| a$ and $d|b$ together imply that $d|(a - q_1 b)$, which is the same as $d|r_1$. Since $b > r_1$, we can divide $b$ by $r1$and apply the division algorithm to obtain:

$$b = q_2 r_1 + r_2 \qquad 0 \leq r_2 < r_1$$

If $r_2 = 0$, then $d = r_1$ and if $r_2 \neq 0$, then $d = \gcd(r_1, r_2)$. The division process continues until some zero remainder appears, say, at the $(n + 1)$th stage where $r_{n-1}$ is divided by $r_n$. The result is the following system of equations:

At each iteration, $d = \gcd(r_i, r_{i+1})$ until finally $d = \gcd(r_n, 0) = r_n$.

Thus, the greatest common divisor of two integers can be found by repetitive application of the division algorithm. This scheme is known as the Euclidean algorithm.

- The first step is to show that $r_n$ divides $a$ and$b$. It follows from the last division in above equation that $r_n$divides $r_{n-1}$.The next to last division shows that $r_n$divides $r_{n-2}$ because it divides both terms on the right.

- Successively, one sees that $r_n$divides all $r_i$"s and finally $a$ and $b$. It remains to show that $r_n$is the largest divisor that divides $a$ and$b$.
- $c$ must divide $r_n$, so that $r_n$= gcd($a$, $b$).

An example with relatively large numbers to see the power of this algorithm:

| To find d=gcd(a,b) = gcd(1160718174,316258250) | | |
|---|---|---|
| $a=q_1b+r_1$ | 1160718174= 3 ×316258250+ 211943424 | d=gcd(316258250,211943424) |
| $b=q_2r_1+r_2$ | 316258250= 1 ×211943424+ 104314826 | d=gcd(211943424, 104314826) |
| $r_1=q_3r_2+r_3$ | 211943424= 2 ×104314826+ 3313772 | d=gcd(104314826, 3313772) |
| $r_2=q_4r_3+r_4$ | 104314826= 31 ×3313772+ 1587894 | d=gcd(3313772, 1587894) |
| $r_3=q_5r_4+r_5$ | 3313772= 2 ×1587894+ 137984 | d=gcd(1587894, 137984) |
| $r_4=q_6r_5+r_6$ | 1587894= 11 ×137984+ 70070 | d=gcd(137984,70070) |
| $r_5=q_7r_6+r_7$ | 137984= 1 ×70070+ 67914 | d=gcd(70070, 67914) |
| $r_6=q_8r_7+r_8$ | 70070= 1 ×67914+ 2156 | d=gcd(67914, 2156) |
| $r_7=q_9r_8+r_9$ | 67914= 31 ×2156+ 1078 | d=gcd(2156, 1078) |
| $r_8=q_{10}r_9+r_{10}$ | 2156= 2 ×1078+ 0 | d=gcd(1078,0) = 1078 |
| Therefore, gcd(1160718174,316258250) = 1078 | | |

- In this example, we begin by dividing 1160718174 by 316258250, which gives 3 with a remainder of 211943424. Next we take 316258250 and divide it by 211943424. The process continues until we get a remainder of 0, yielding a result of 1078.
  - For every step of the iteration, we have $r_{i-2} = q_i r_{i-1} + r_i$, where $r_{i-2}$ is the dividend, $r_{i-1}$ is the divisor, $q_i$ is the quotient, and $r_i$ is the remainder.

## MODULAR ARITHMETIC

**Explain the Modular Arithmetic operation and properties in detail.**

**The Modulus**

- If $a$ is an integer and $n$ is a positive integer, we define $a$ mod $n$ to be the remainder when $a$ is divided by $n$. The integer $n$ is called the **modulus**. Thus, for any integer $a$, we can write follows:

| 11 mod 7 = 4; | -11 mod 7 = 3 |
|---|---|

- Two integers $a$ and $b$ re said to be congruent modulo $n$, if ($a$ mod $n$) = ($b$ mod $n$). This is written as $a \equiv b \pmod{n}$[2].

| | |
|---|---|
| 73 = 4 (mod 23); | 21 = 19 (mod 10) |

If $a \equiv 0 \pmod{n}$, then $n|a$.

**Properties of Congruence**

congruence have the following properties:

**1.** $a \equiv b \pmod{n}$ if $n|(a - b)$.

**2.** $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$.

**3.** $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

First property:

• Define ($a$ mod $n$) = $r_a$and ($b$ mod $n$) = $r_b$. Then we can write $a = r_a + jn$for some integer $j$ and $b = r_b + kn$for some integer $k$.

Then

$(a + b) \bmod n = (ra + jn + rb + kn) \bmod n = (ra + rb + (k + j)n) \bmod n$

$= (ra + rb) \bmod n$

$= [(a \bmod n) + (b \bmod n)] \bmod n$

• Examples of the three properties:

11 mod 8 = 3; 15 mod 8 = 7

[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 =2

(11 + 15) mod 8 =26 mod 8 =2

[(11 mod 8) – (15 mod 8)] mod 8 = -4 mod 8 = 4

(11-15) mod 8 = -4 mod 8 = 4

[(11 mod 8)× (15 mod 8)] mod 8 = 21 mod 8 =5

(11 × 15) mod 8 = 165 mod 8 =5

• Exponentiation is performed by repeated multiplication, as in ordinary arithmetic.

To find $11^7$ mod 13,

$11^2 = 121 = 4 \pmod{13}$

$11^4 = (11^2)^2 = 4^2 = 3 \pmod{13}$

$11^7 = 11 \times 4 \times 3 = 132 = 2 \pmod{13}$

Thus, the rules for ordinary arithmetic involving addition, subtraction, and multiplication carry over into modular arithmetic. The following table below provides an illustration of modular addition and multiplication modulo 8.multiplication carry over into modular arithmetic. The following table below provides an illustration of modular addition and multiplication modulo 8.

• Both matrices are symmetric about the main diagonal in conformance to the commutative property of addition and multiplication.

• As in ordinary addition, there is an additive inverse, or negative, to each integer in modular arithmetic. In this case, the negative of an integer $x$ is the integer $y$ such that $(x + y) \bmod 8 = 0$.

• To find the additive inverse of an integer in the left-hand column, scan across the corresponding row of the matrix to find the value 0; the integer at the top of that column is the additive inverse; thus, (2 + 6) mod 8 = 0.Similarly, the entries in the multiplication table are straightforward.

• In modular arithmetic mod 8, the multiplicative inverse of $x$ is the integer $y$ such that $(x\ y) \bmod 8 = 1 \bmod 8$.

**Table: Arithmetic Modulo 8**

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 |
| 3 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 |
| 4 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| 5 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 |
| 6 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 |
| 7 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

(a) Addition modulo 8

| × | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 0 | 2 | 4 | 6 | 0 | 2 | 4 | 6 |
| 3 | 0 | 3 | 6 | 1 | 4 | 7 | 2 | 5 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 0 | 4 |
| 5 | 0 | 5 | 2 | 7 | 4 | 1 | 6 | 3 |
| 6 | 0 | 6 | 4 | 2 | 0 | 6 | 4 | 2 |
| 7 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

(b) Multiplication modulo 8

| $w$ | $-w$ | $w^{-1}$ |
|---|---|---|
| 0 | 0 | — |
| 1 | 7 | 1 |
| 2 | 6 | — |
| 3 | 5 | 3 |
| 4 | 4 | — |
| 5 | 3 | 5 |
| 6 | 2 | — |
| 7 | 1 | 7 |

(c) Additive and multiplicative inverse modulo 8

**Properties of Modular Arithmetic**

Define the set $Z_n$ as the set of nonnegative integers less than $n$:

$Z_n = \{0, 1, \ldots\ldots, (n - 1)\}$

This is referred to as the **set of residues**, or **residue classes** (mod $n$). To be more precise, each integer in $Z_n$ represents a residue class. We can label the residue classes (mod $n$) as [0], [1], [2], ….., [$n$ - 1], where

$[r] = \{a: a \text{ is an integer}, a \equiv r \pmod{n}\}$

The residue classes (mod 4) are

[0] = {…,-16,-12,-8,-4,0,4,8,12,16,…}

[1] = {…,-15,-11,-7,-3,1,5,9,13,17,…}

[2] = {…,-14,-10,-6,-2,2,6,10,14,18,…}

[3] = {…,-13,-9,-5,-1,3,7,11,15,19,…}

• Of all the integers in a residue class, the smallest nonnegative integer is the one used to represent the residue class. Finding the smallest nonnegative integer to which $k$ is congruent modulo $n$ is called **reducing *k* modulo *n***. $Z_n$ is a commutative ring with a multiplicative identity element.

**Property Expression**

**Commutative Laws** $(w+x) \bmod n = (x+w) \bmod n$

$(w \times x) \bmod n = (x \times w) \bmod n$

**Associative Laws** $[(w+x)+y] \bmod n = [w+(x+y)] \bmod n$

$[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$

**Distributive Laws** $[w \times (x+y)] \bmod n = [(w \times x)+(w \times y)] \bmod n$

**Identities** $(0+w) \bmod n = w \bmod n$

$(1 \times w) \bmod n = w \bmod n$

**Additive Inverse** (-w) For each $w \in Z_n$, there exists a z such that $w+z=0 \bmod n$

There is one peculiarity of modular arithmetic that sets it apart from ordinary arithmetic.**if** $(a + b) \equiv (a + c) \pmod{n}$ **then** $b \equiv c \pmod{n}$

$(5 + 23) \equiv (5 + 7) \pmod 8$; $23 \equiv 7 \pmod 8$

• Equation above is consistent with the existence of an additive inverse. Adding the additive inverse of $a$ to both sides of Equation above, we have

$((-a) + a + b) \equiv ((-a) + a + c) \pmod{n} b \equiv c \pmod{n}$

• However, the following statement is true only with the attached condition:

**if**($a\ b$) ($a\ c$)(mod $n$) **then** $b\ c$ (mod $n$) **if** $a$ is relatively prime to $n$

• Two integers are **relatively prime** if their only common positive integer factor is 1.

• Applying the multiplicative inverse of $a$ to both sides, we have

$((a^{-1})ab)$ $((a^{-1})ac)$ (mod$n$)

$b\ c$ (mod $n$)

• In general, an integer has a multiplicative inverse in $Z_n$if that integer is relatively primeto $n$.

## Discrete Logarithms

The inverse problem to exponentiation is to find the discrete logarithm of a number modulo p •

find x such that y = gx (mod p)

$x = \log_g y \pmod p$

if g is a primitive root then it always exists, otherwise it may not, eg. $x = \log_3 4$ mod 13 has no answer $x = \log_2 3 \bmod 13 = 4$ by trying successive powers • whilst exponentiation is relatively easy, finding discrete logarithms is generally a hard problem