UNIT - I:
Basic Principles
Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography

Ch-1 - Introduction

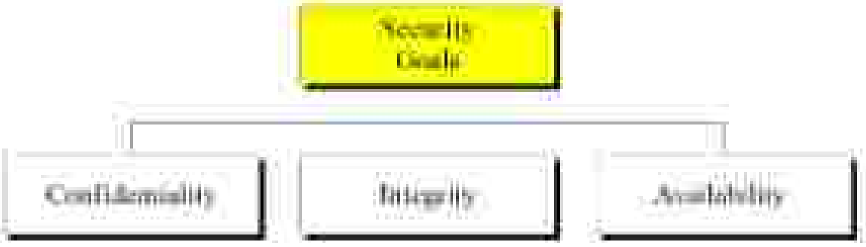
The generic name for the collection of tools designed to protect data and to thwart (prevent from succeeding in) hackers is **computer security**.

The term **network security** is somewhat misleading, because virtually all business government and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet, and the term **internet security** is used.

There are no clear boundaries between these two forms of security. For example, one of the most publicized types of attack on information systems is the computer virus. A virus may be introduced into a system physically when it arrives in a diskette and is subsequently loaded onto a computer, viruses may also arrive over an internet, in either case, once the virus is resident on a computer system, internal computer security tools are needed to detect and recover from the virus.

1.1 Security Goals

Let us first discuss the three security goals: confidentiality, integrity, and availability



Confidentiality:

- It is the most common aspect of information security. We need to protect our confidential information.
- An organization needs to guard against those malicious actions that endanger the confidentiality of its information.
- In the military, concealment of sensitive information is the major concern.
- In industry, hiding some information from competitors is crucial to the operation of the organization.
- In banking, customer's accounts need to be kept secret.
- Confidentiality not only applies to the storage of the information, is also applies to the transmission of information.

Integrity:

- Information needs to be changed constantly. In a bank, when a customer deposits or withdraws money, the balance of her account needs to be changed.

- Integrity means that changes need to be done only by authorised entities and through authorised mechanisms.
- Integrity violation is not necessarily the result of a malicious act: an interruption in the system, such as a power surge (up and down), may also create unwanted changes in some information.

Availability:

- The third component of information security is availability.
- The information created and stored by an organization needs to be available to authorised entities. Information is useless, if it is not available.
- The unavailability of information is just as harmful for an organization as the lack of confidentiality or integrity.

Before beginning further topics, we define some terms:

- An original message is known as the **plaintext**, while the coded message is called the **ciphertext**.
- The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**.
- Restoring the plaintext from the ciphertext is **deciphering** or **decryption**.
- The many schemes used for enciphering constitute the area of study known as **cryptography**.
- Such a scheme is known as a **cryptographic system** or a **cipher**.
- Technology used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**.
- Cryptanalysis is what the layperson calls "breaking the code". The areas of cryptography and cryptanalysis together are called **cryptology**.

Symmetric cipher model

A symmetric encryption scheme has five ingredients (Figure 1.1):

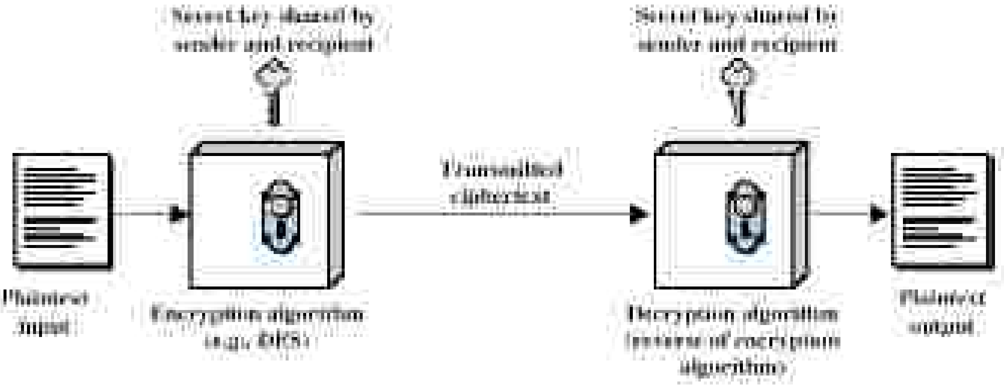

Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 1.1 Simplified Model of Conventional Encryption

- **Plaintext**: This is the original intelligible message of data that is fed into the algorithm as input.
- **Encryption algorithm**: the encryption algorithm performs various substitutions and transformations on the plaintext.

- **Secret key**: the secret key is also input to the encryption algorithm. The key is a value independent of the plaintext. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext**: this is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm**: this is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

There are two requirements for secure use of conventional encryption:

1) We need a strong encryption algorithm.
2) Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.

With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext Y.

We can write this as

$$Y = E_K(X)$$

This notation indicates that ciphertext Y is from plaintext X using encryption algorithm E and secret key K.

The intended receiver, in possession of the key, is able to invert the transformation:

$$X = D_K\ (Y)$$

Plaintext X is produced by using decryption algorithm D and secret key K.

1.2 Cryptographic Attacks

Cryptographic attacks can be broadly categorized into two distinct types: 1. Cryptanalytic and 2. Non-cryptanalytic.

❑ **Cryptanalytic attacks**: These attacks are combination of statistical and algebraic techniques aimed at ascertaining the secret key of a cipher.

- These methods inspect the mathematical properties of the cryptographic algorithms and aims at finding distinguishers of the output distribution of cryptographic algorithms form uniform distributions.
- The objective of cryptanalysis is to find properties of the cipher which does not exist in a random function.
- Here distinguishers means that all attacks are fundamentally distinguishers. The attacker thus guesses the key and looks for the distinguishing property. If the property is detected, the guess is correct otherwise the next guess is tried.
- The guessing complexity is lesser than the brute force search complexity.

❑ **Non-cryptanalytic attacks**:

- The other types of attacks are non-cryptanalytic attacks, which do not exploit the mathematical weakness of the cryptographic algorithm.

The three goals of security — confidentiality, integrity, and availability---can be very much threatened by this class of attacks.

Although the literature uses different approaches to categorizing the attacks, we will first divide them into three groups related to the security goals. Later, we will divided them into two broad categories based on their effects on the system. Figure 1.2 shows the first taxonomy.

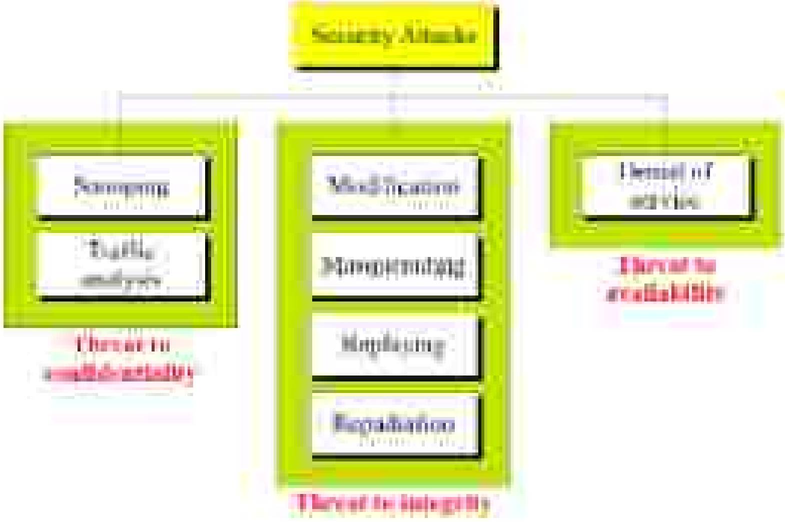
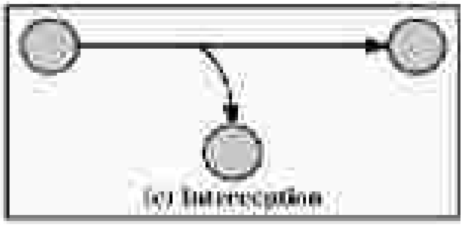


Figure 1.2 Taxonomy of attacks with relation to security goals

❑ **Attacks threatening confidentiality**: In general , two types of attacks threaten the confidentiality of information: snooping and traffic analysis.

Snooping:

- It refers to unauthorized access to (or) ***interception of data***. For example, a file transferred through the internet may contain confidential information.
- An unauthorized entity may interrupt the transmission and use the contents for her own benefit.
- To prevent snooping, the data can be made non-intelligible to the intercepter by using encipherment techniques.



Traffic analysis:

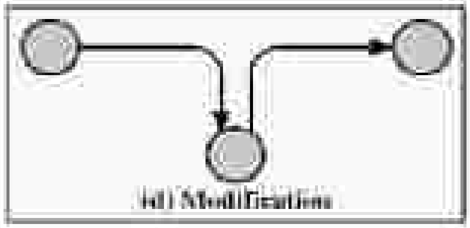
- Although encipherment of data may it non intelligible for the interceptor, she can obtain some other type information by monitoring online traffic.
- While passing the data through networks the opponents may observe the route of the data and amount of data and the passing time of data.

❑ **Attacks threatening integrity**:

The integrity of data can be threatened by several kinds of attacks: modification, masquerading, replaying and repudiation.

Modification:

- After intercepting or accessing information, the attacker modifies the information to make it beneficial to herself.
- For example, a customer sends a message to a bank to do some transaction. The attacker intercepts the message and changes the type of transaction to benefit herself.
- Note that sometimes the attacker simply deletes or delays the message to harm the system or to benefit from it.



(d) Modification

Masquerading:

- It happens when the attacker impersonates somebody else.
- For example, an attacker might steal the bank card PIN of a bank customer and pretend that she is that customer.
- Another example, a user tries to contact a bank, but another site pretends that it is the bank and obtains some information from the user.

Replaying:

- The attacker obtains a copy of a message sent by a user and later tries to replay it.
- For example, a person sends a request to her bank to ask for payment to the attacker, who has done a job for her. The attacker intercepts the message and sends it again to receive another payment from the bank.

Repudiation:

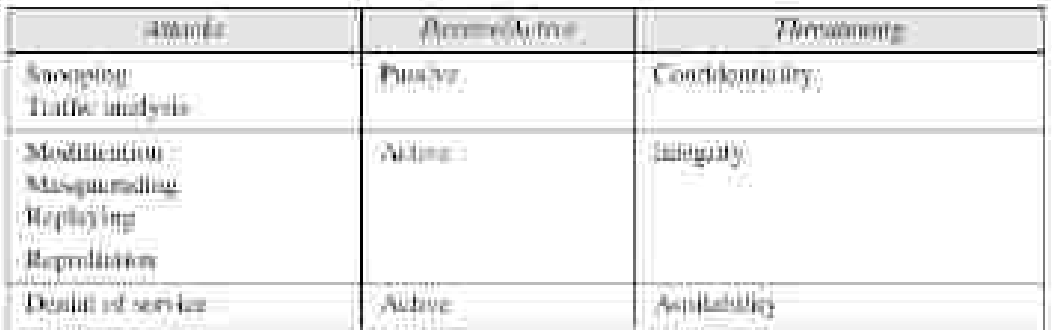
- This type of attack is different from others because it is performed by one of the two parties in the communication: the sender or the receiver.
- The sender of the message might later deny that she has sent the message; the receiver of the message might later deny that she has received the message.
- An example of denial by the sender would be a bank customer asking her bank to send some money to a third party but later denying that she has made such a request.
- An example of denial by the receiver could occur when a person buys a product from a manufacturer and pays for it electronically, but the manufacturer later denies having received the payment and asks to be paid.

❑ **Attacks Threatening Availability**:

Denial of Service:

- Denial of Service (DoS) is a very common attack. It may slow down or totally interrupt the service of a system.
- The attacker can use several strategies to achieve this. She might send so many bogus requests to a server that the server crashes because of heavy load.

- **Passive verses Active Attacks**



Attacks	Passive/Active	Threatening
Snooping Traffic analysis	Passive	Confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of service	Active	Availability

Table 1.1 Categorization of passive and active attacks

Passive attacks:

- In a passive attack, attacker's goal is just to obtain information. This means that the attack does not modify data or harm the system.
- The system continues with its normal operation. However the attack may harm the sender or the receiver of the message.
- Attacks that threaten confidentiality – snooping and traffic analysis – are passive attacks.
- Passive attacks, however, can be prevented by encipherment of the data.

Active attacks:

- An active attack may change the data or harm the system.
- Attacks that threaten the integrity and availability are active attacks.
- These attacks are normally easier to detect than to prevent because an attacker can launch them in a variety of ways.

1.3 Services and Mechanism

ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) provides some security services and some mechanisms to implement those services.

1.3.1 Security Services

ITU-T (X.800) has defined five services related to the security goals and attacks we defined in the previous sections. Figure 1.3 shows the taxonomy of those five common services.

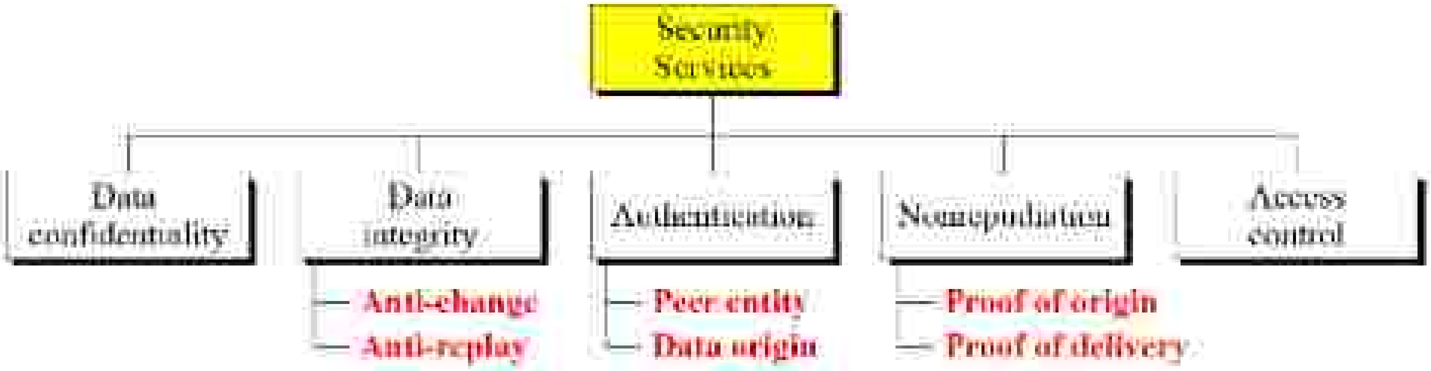


Figure 1.3 Security Services

It is easy to relate one or more of these services to one or more of the security goals. It is also easy to see that these services have been designed to prevent the security attacks that we have mentioned.

Data Confidentiality: Data confidentiality is designed for information is not made available to unauthorised individual. It is designed to prevent snooping and traffic analysis attacks.

Data Integrity: It is designed to protect data from modification, insertion, deletion, and replaying by an adversary. It may protect the whole message or part of the message.

Authentication: This service provides the authentication of the party at the other end of the line. In connection oriented communication, it provides authentication of the sender and receiver during the connection establishment. In connection-less communication, it authenticates the source of the data (data origin authentication).

Nonrepudiation: Nonrepudiation service protects against repudiation (refuse to accept) by either the sender or the receiver of the data.

> **Nonrepudiation, Origin**
> Proof that the message was sent by the specified party.
> **Nonrepudiation, Destination**
> Proof that the message was received by the specified party.

Access Control: It provides protection against unauthorised access to data. The term access in this definition is very broad and can involve reading, writing, modifying, executing programs, and so on.

1.3.2 Security Mechanisms

ITU-T (X.800) also recommends some security mechanisms to provide the security services defined in the previous section. Figure 1.4 gives the taxonomy of these mechanisms.

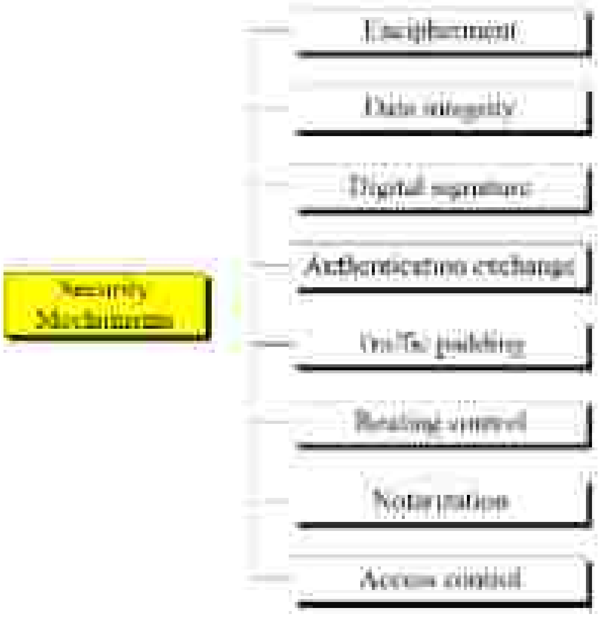


Figure 1.4 Security Mechanisms

Encipherment:

- Encipherment, hiding or covering data, can provide confidentiality.
- Today two techniques cryptography and steganography are used or enciphering.

Data integrity:

- The data integrity mechanism appends to the data a short checkvalue that has been created by a specific process from the data itself.
- The receiver receives the data and checks value.
- He creates a new checkvalue from the received data and compares the newly created checkvalue with the one received.
- If two check values are same, the integrity of data has been preserved.

Digital signature:

- A digital signature is a means by which the sender can electronically sign the data and receiver can electronically verify the signature.
- The sender uses a process that involves showing that she owns a private key related to the public key that she has announced publicly.
- The receiver uses the sender's public key to prove that the message is indeed signed by the sender who claims to have sent the message.

Authentication exchange:

- In this two entities exchange some messages to prove their identity to each other.
- For example, one entity can prove that she knows a secret that only she is supposed to know

Traffic Padding:

- This means inserting some bogus data into the data traffic to the adversary's attempt to use the traffic analysis.

Routing control:

- It means selecting and continuously changing different available routes between sender and receiver to prevent the opponent from eavesdropping (secretly listen to a conversation) on a particular route.

Notarization:

- It means selecting a third trusted party to control the communication between two entities.
- This can be done, for example, to prevent repudiation.

Access control:

- It uses methods to prove that a user has access right to the data or resources owned by a system.
- Examples of proofs are passwords and PINs.

1.3.3 Relation between Services and Mechanisms

Table 1.2 shows the relationship between the security services and the security mechanisms.

Security Service	*Security Mechanism*
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

Table 1.2 Relation between security services and security mechanisms

Ch-2 - Mathematics of Cryptography:
(Modular Arithmetic, Congruence, and Matrices)

Cryptography is based on some specific areas of mathematics, including number theory, linear algebra and algebraic structures.

2.1 Integer arithmetic

In integer arithmetic, we use a set and few operations.

Set of Integers: The set of integers, denoted by Z, contains all integral numbers (with no fraction) from negative infinity to positive infinity. (Figure 2.1)

$$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

Figure 2.1 The set of integers

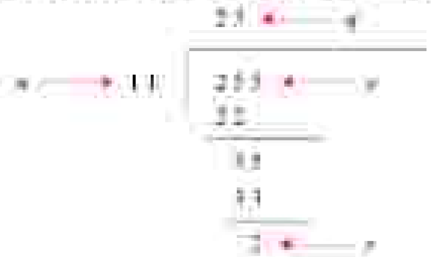
2.1.1 Integer Division

In integer arithmetic, if we divide a by n, we get q and r. The relationship between these four integers can be shown as

$$a = q \times n + r$$

In this relation, *a* is called the *dividend*; *q*, the *quotient*; *n*, the *divisor*; and *r*, the *remainder*.

Example:
Assume that a=255 and n=11. We can find q=23 and R=2 using the division algorithm.



Two Restrictions: For our purpose, we impose two restrictions. First, we require that the divisor be a positive integer (n>0). Second, we require that the remainder be a non-negative integer(r ≥ 0).

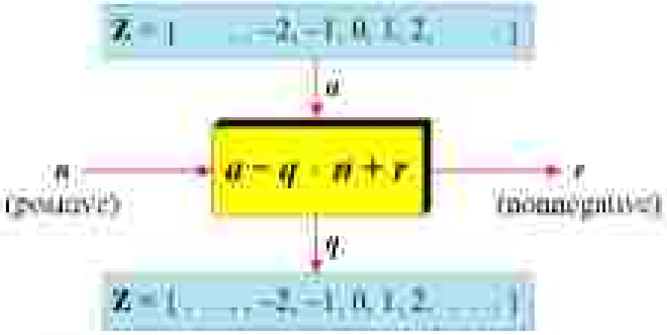

Z = {..., −2, −1, 0, 1, 2, ...}
a
n
(positive)
a = q × n + r
r
(nonnegative)
q
Z = {..., −2, −1, 0, 1, 2, ...}

Figure 2.2 Division algorithm for integers

2.1.2 Divisibility

If a is not zero and we let r = 0 in the division relation, we get

$$a = q \times n$$

We say that *n* divides *a*. we can also say that *a* is divisible by *n*. when we are not interested in the value of *q*, we can write the above relationship as a|n. If the remainder is not zero, then *n* does not divide *a* and we can write the relationship as a ∤ n.

Properties:

- If a|1, then a = ±1.
- If a|b and b|a, then a = ±b.
- If a | b and b | c, then a | c
- If a|b and a|c, then a | (m×b + n×c) where m and n are arbitrary integers.

Example:

a. Since 3|15 and 15|45, according to third property, 3|45

b. Since 3|15 and 3|9, according to fourth property, 3| (15×2+9×4), which means 3|66

Greatest Common Divisor (gcd): One integer often needed in cryptography is the greatest common divisor of two positive integers. Two positive integers may have many common divisors, but only one greatest common divisor.

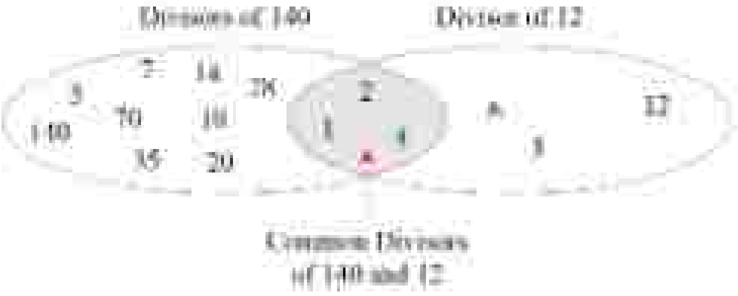


Figure 2.3 Common divisors of two integers

> **Note**: The greatest common divisor of two positive integers is the largest integer that can divide both integers

Euclidean Algorithm:

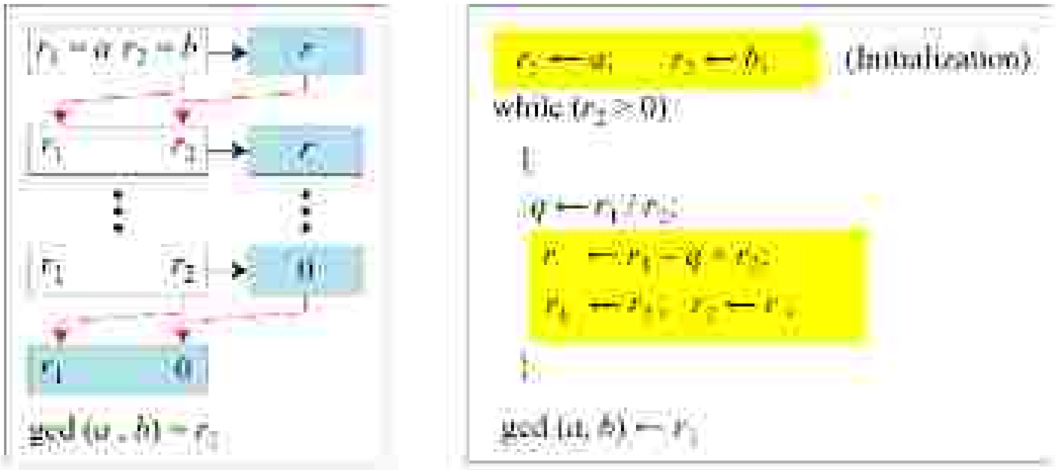
Finding the greatest common divisor (gcd) of two positive integers by listing all common divisors is not practical when two integers are large.

The Euclidean algorithm is based on the following two facts:

> Fact 1: gcd (a, 0) = a
> Fact 2: gcd (a, b) = gcd (b, r), where *r* is the remainder of dividing a by b

The first fact tells us that if the second integer is 0, the greatest common divisor is the first one. The second fact allows us to change the value of a, b until b becomes 0.

> gcd (36, 10) = gcd(10, 6) = gcd(6, 4) = gcd(4, 2) = gcd(2, 0) = 2



```
$r_1 \leftarrow a;$    $r_2 \leftarrow b;$    (Initialization)
while ($r_2 > 0$)
{
    $q \leftarrow r_1 / r_2;$
    $r \leftarrow r_1 - q \times r_2;$
    $r_1 \leftarrow r_2;$  $r_2 \leftarrow r;$
}
gcd $(a, b) \leftarrow r_1$
```

a. Process   b. Algorithm

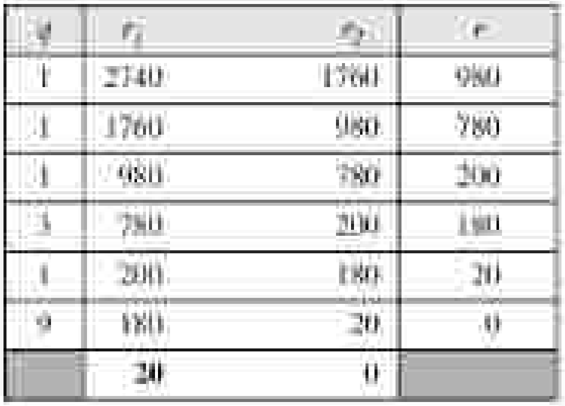
Figure 2.4 Euclidean Algorithm

- We use two variables r1 and r2, to hold the changing values during the process of reduction. They are initialized to a and b.
- In each step, we calculate the remainder of r1 divided by r2 and store the result in the variable r. we then replace r1 by r2 and r2 by r.
- The steps are continued until r2 becomes 0. At this moment, we stop. The gcd (a, b) is r1.

When gcd (a, b) = 1, we say that a and b are ***relatively prime***

Example: Find the greatest common divisor of 2740, 1760

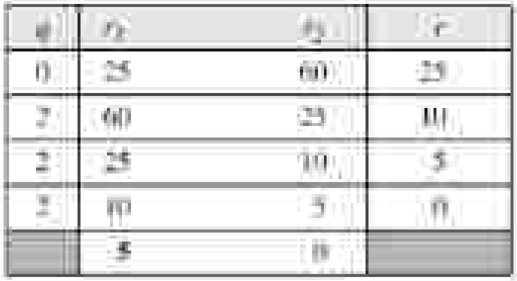
Solution: We have gcd (2740, 1760) = 20.



q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Example: Find the greatest common divisor of 25 and 60.

Solution: We have gcd (25, 60) = 5.



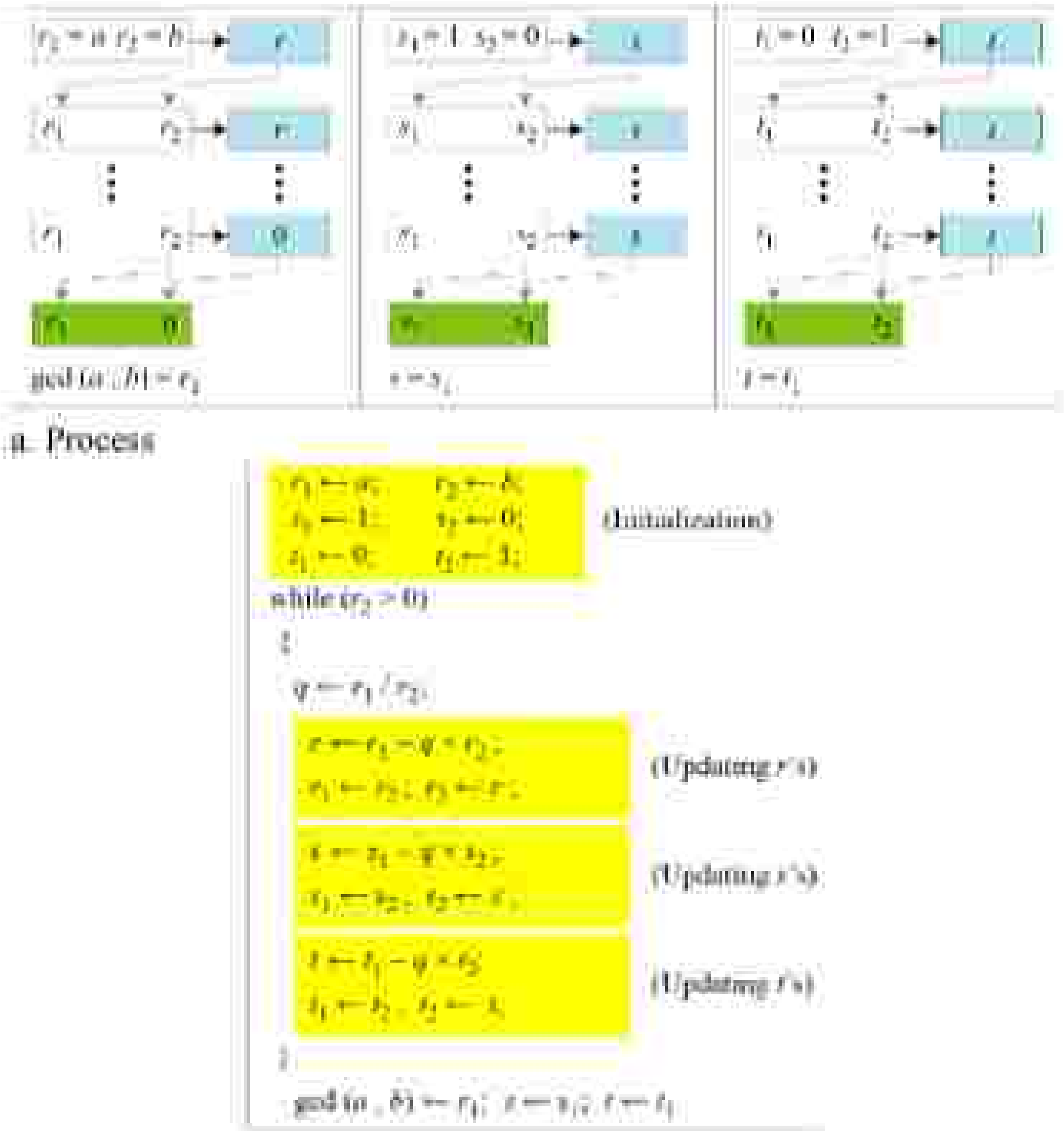
q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

2.2 The Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a,\ b)$$

The **Extended Euclidean algorithm** can calculate the gcd (a, b) and at the same time calculate the value of s and t. The algorithm and the process is shown below diagram.



a. Process

```
$r_1 \leftarrow a;$        $r_2 \leftarrow b;$
$s_1 \leftarrow 1;$        $s_2 \leftarrow 0;$        (Initialization)
$t_1 \leftarrow 0;$        $t_2 \leftarrow 1;$
while ($r_2 > 0$)
{
    $q \leftarrow r_1 / r_2;$

    $r \leftarrow r_1 - q \times r_2;$
    $r_1 \leftarrow r_2;\ r_2 \leftarrow r;$        (Updating r's)

    $s \leftarrow s_1 - q \times s_2;$
    $s_1 \leftarrow s_2;\ s_2 \leftarrow s;$        (Updating s's)

    $t \leftarrow t_1 - q \times t_2;$
    $t_1 \leftarrow t_2;\ t_2 \leftarrow t;$        (Updating t's)
}
    $\gcd(a, b) \leftarrow r_1;\ s \leftarrow s_1;\ t \leftarrow t_1$
```

b. Algorithm

Figure 2.5 Extended Euclidian Algorithm

Example: Using the extended Euclidean algorithm, find the greatest common divisor of the following pairs and the value of s and *t*.

a) 161 and 28 b) 4 and 7 c) 291 and 42

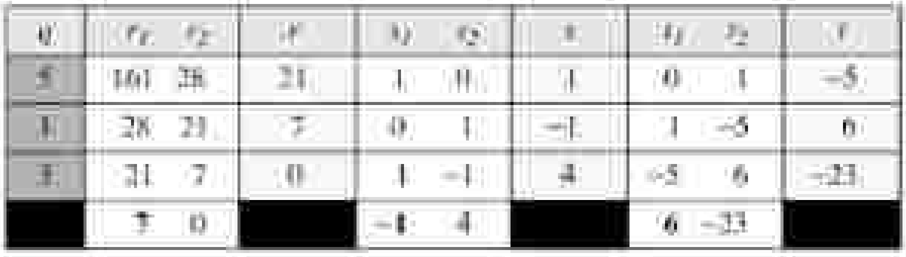
d) 84and 320 e) 400 and 60 f) 17 and 0

g) 0 and 45

Solution to a):

We get gcd (161, 28) = 7 ,s = −1 and t = 6.

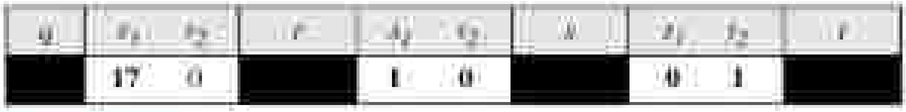
$r = r_1 - q \times r_2 \quad s = s_1 - q \times s_2 \quad t = t_1 - q \times t_2$



q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	−5
1	28	21	7	0	1	−1	1	−5	6
3	21	7	0	1	−1	4	−5	6	−23
	7	0		−1	4		6	−23	

Solution to f)

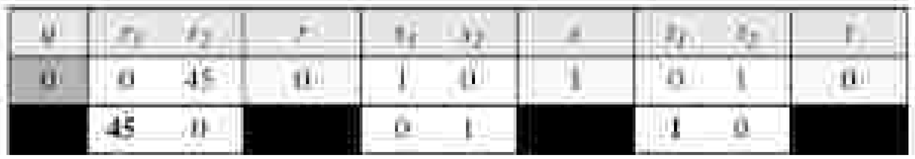
We get gcd (17, 0) = 17, s = 1, and t = 0.



q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	17	0		1	0		0	1	

Solution to g)

We get gcd (0, 45) = 45, s = 0, and t = 1.



q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
0	0	45	0	1	0	1	0	1	0
	45	0		0	1		1	0	

2.2.1 Linear Diophantine Equations

Although we will see a very important application of the extended Euclidean algorithm. One immediate applications is to find the solutions to the linear Diophantine equations of two variables, an equation of type ax + by = c. We need to find integer values for x and y that satisfy the equation. This type of equation has *either no solution or an infinite number of solutions*.

Let d= gcd (a, b). If d ∤ c, then the equation has no solution.
If d | c, then we have an infinite number of solutions. One of them is called the particular; the rest, general.

A linear Diophantine equation of two variables is $ax + by = c$.

Particular Solution If d | c, a particular solution to the above equation can be found using the following steps:

1. Reduce the equation to $a_1x + b_1y = c_1$ by dividing both sides of the equation by d. this is possible because d divides a, b, and c by the assumption.
2. Solve for s and t in the relation $a_1s + b_1y = 1$ using the extended Euclidean algorithm.
3. The particular solution can be found:

Particular Solution: $x_0 = (c/d)\,s$ and $y_0 = (c/d)\,t$

General Solutions After finding the particular solution, the general solutions can be found:

General Solutions: $x = x_0 + k(b/d)$ and $y = y_0 - k(a/d)$

Example:

Find the particular and general solutions to the equation 40x + 16y = 88.

Solution:

The equation is of the form ax + by = c

Step1:

Find d = gcd(a, b)
= gcd(40, 60)
= gcd(16, 8)
= gcd(8, 0)
= 8

Step2:

d | c so,

Divide both sides of the equation by d

$$\frac{40x}{8} + \frac{16y}{8} = \frac{88}{8}$$

$\Rightarrow 5x + 2y = 11$

Step3:

Solve for s and t in the relation 5x + 2y = 1 using extended Euclidean algorithm.

q	r1	r2	r	s1	s2	s	t1	t2	t
2	5	2	1	1	0	1	0	1	-2
2	2	1	0	0	1	-2	1	-2	5
	1	0		1	-2		-2	5	

s = 1, t = -2

Particular solution: $x_0 = (c/d)s$ $\quad$ $y_0 = (c/d)t$

$= (88/8)*1$ $\quad$ $= (88/8) * (-2)$

$= 11$ $\quad$ $= -22$

General solutions: $x = x_0 + k(b/d)$ $\quad$ $y = y_0 - k(a/d)$ Where k is an integer

$x = 13, \ldots$ $\quad$ $y = -27, \ldots$

Exercise:

Find the particular and general solutions to the following linear Diophantine equations.

a) 25x + 10y = 15 $\quad$ b) 19x + 13y = 20 $\quad$ c) 14x + 21y = 77 $\quad$ d) 40x + 16y = 88

2.3 Modular Arithmetic

- The division relationship ($a=q \times n + r$) has two inputs (a and n) and two outputs (q and r).
- In modular arithmetic, we are interested in only one of the outputs, the remainder r. We don't care about the quotient q.
- In other words, we want to know what is the value of r when we divide a by n.
- This implies that we can change the above relation into a binary operator with two inputs a and n and one output r.

2.3.1 Modulo Operator

- The above mentioned binary operator is called the **modulo operator** and is shown as **mod**.
- The second input (n) is called the **modulus**. The output r is called the **residue**.
- The below figure shows, the modulo operator (mod) takes an integer (a) from the set z and a positive modulus (n) .The operator creates a nonnegative residue (r). We can say

a mod n = r

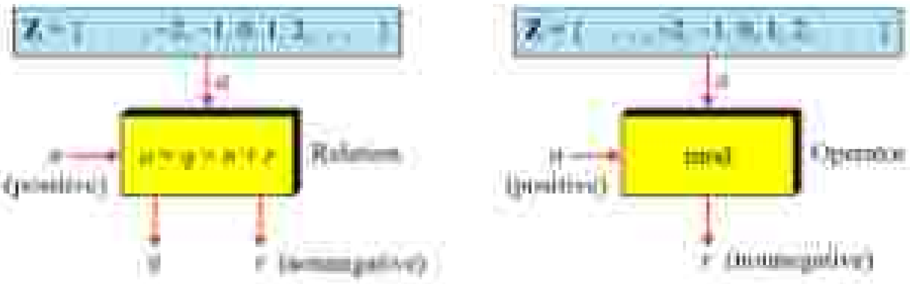


Figure 2.6 Division relation and modulo operator

EXAMPLE Find the results of the following operations:

a. 27 mod 5
b. 36 mod 12
c. −18 mod 14
d. −7 mod 10

SOLUTION We are looking for the residue r. We can divide the a by n and find q and r. We can then disregard q and keep r.

a. Dividing 27 by 5 results in $r = 2$. This means that 27 mod 5 = 2.
b. Dividing 36 by 12 results in $r = 0$. This means that 36 mod 12 = 0.
c. Dividing −18 by 14 results in $r = -4$. However, we need to add the modulus (14) to make it nonnegative. We have $r = -4 + 14 = 10$. This means that −18 mod 14 = 10.
d. Dividing −7 by 10 results in $r = -7$. After adding the modulus to −7, we have $r = 3$. This means that −7 mod 10 = 3.

2.3.2 Set of Residues: Z_n

- The result of modulo operation with modulus n is always an integer between 0 and n-1.
- In other words, the result of a mod n is always a nonnegative integer less than n.
- We can say that modulo operation creates a set, which in modular arithmetic is referred to as the **set of least residues modulo n**, or $\mathbf{Z_n}$.
- We have infinite instances of the set of residues (Z_n), one for each value of n.
- The below figure shows the set Z_n and three instances, Z_2, Z_6, and Z_{11}.

$$\mathbf{Z}_n = \{0, 1, 2, 3, \ldots, (n-1)\}$$

$$\mathbf{Z}_2 = \{0, 1\} \quad \mathbf{Z}_6 = \{0, 1, 2, 3, 4, 5\} \quad \mathbf{Z}_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Figure 2.7 Some Z_n sets

2.3.3 Congruence

- In Cryptography, we often used the concept of congruence instead of equality.
- Mapping from Z to Z_n is not one-to-one.
- For example, the result of 2 mod 10 = 2, 12 mod 10 = 2, 22 mod 10 = 2, and so on.
- In Modular arithmetic, integers like **2, 12, and 22 are called congruent mod 10**.
- To show that two integers congruent, we use the **congruence operator (≡)**.
- We add the phrase (mod n) to the right side of the congruence to define the value of modulus that makes the relationship valid. For example ,we write:

$$2 \equiv 12 \pmod{10} \qquad 13 \equiv 23 \pmod{10} \qquad 34 \equiv 24 \pmod{10} \qquad -8 \equiv 12 \pmod{10}$$
$$3 \equiv 8 \pmod{5} \qquad 8 \equiv 13 \pmod{5} \qquad 23 \equiv 33 \pmod{5} \qquad -8 \equiv 2 \pmod{5}$$

2 mod 10 = 12 mod 10 — 2 ≡ 12 mod 12

We need to explain several points.

- The congruence operator looks like the equality operator, but there are differences. First, an equality operator maps a member of Z to itself; the congruence operator maps a member from Z to member of Z_n. Second, the equality operator is one-to-one; the congruence operator is many-to-one.
- The phrase (mod n) that we insert at the right-hand-side of the congruence operator is just an indication of the destination set (Z_n).

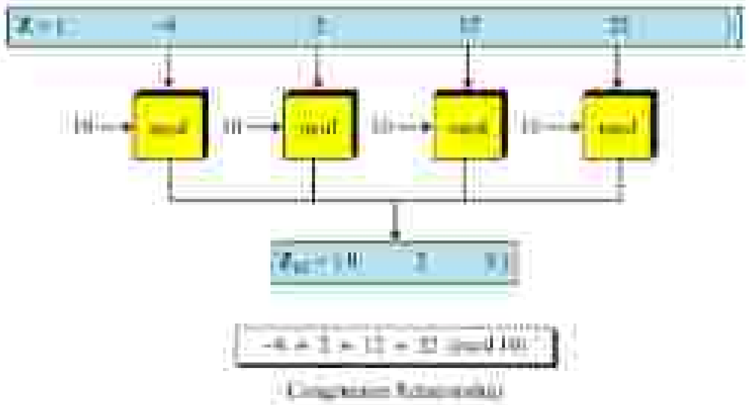
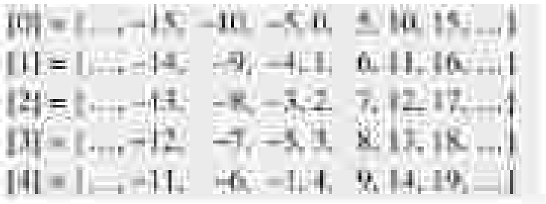


Figure 2.8 Concept of congruence

Residue classes

- A residue calss [a] or $[a]_n$ is the set of integers congruent modulo n. In other words, it is the set of all integers such that x = a (mod n). For example, if n=5, we have five sets [0], [1], [2], [3], and [4] as shown below:

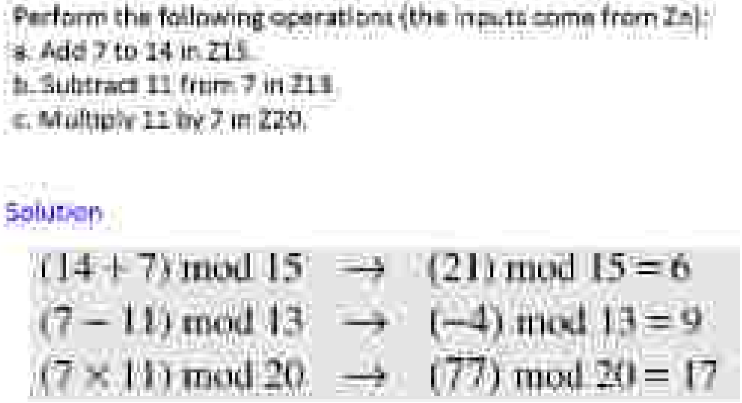


$$
\begin{aligned}
[0] &= \{\ldots, -15, -10, -5, 0, 5, 10, 15, \ldots\}\\
[1] &= \{\ldots, -14, -9, -4, 1, 6, 11, 16, \ldots\}\\
[2] &= \{\ldots, -13, -8, -3, 2, 7, 12, 17, \ldots\}\\
[3] &= \{\ldots, -12, -7, -2, 3, 8, 13, 18, \ldots\}\\
[4] &= \{\ldots, -11, -6, -1, 4, 9, 14, 19, \ldots\}
\end{aligned}
$$

- The integers in the set [0] are all reduced to 0 when we apply modulo 5 operation on them. The integers in the set [1] are all reduced to 1 when we apply modulo 5 operation, and so on.
- In each set, there is one element called the least (non-negative) residue.
- In the set [0], this element is 0; in the set [1], this element is 1; and so on.
- The set of all of these least residues is what we have shown as $Z_5 = \{0, 1, 2, 3, 4\}$.
- In other words, the set Z_n is the set of all least residue modulo n.

2.3.4 Operations in Z_n:

Example

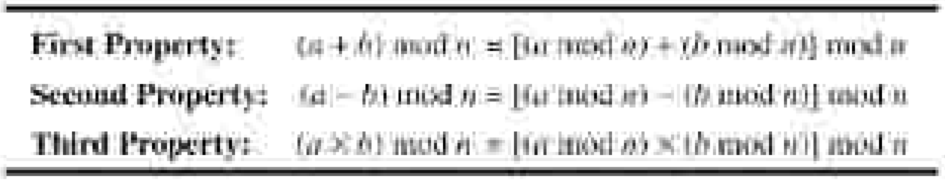


Perform the following operations (the inputs come from Zn):
a. Add 7 to 14 in Z15.
b. Subtract 11 from 7 in Z13.
c. Multiply 11 by 7 in Z20.

Solution

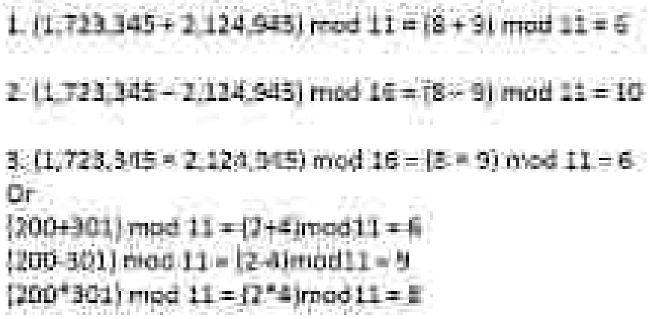
$$
\begin{aligned}
(14 + 7) \bmod 15 &\rightarrow (21) \bmod 15 = 6\\
(7 - 11) \bmod 13 &\rightarrow (-4) \bmod 13 = 9\\
(7 \times 11) \bmod 20 &\rightarrow (77) \bmod 20 = 17
\end{aligned}
$$

Properties of mod operator



First Property:	$(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$
Second Property:	$(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$
Third Property:	$(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

The following are shows the applications of the above properties



1. (1,723,345 + 2,124,945) mod 11 = (8 + 9) mod 11 = 6

2. (1,723,345 − 2,124,945) mod 16 = (8 − 9) mod 11 = 10

3. (1,723,345 × 2,124,945) mod 16 = (8 × 9) mod 11 = 6
Or
(200+301) mod 11 = (2+4)mod11 = 6
(200-301) mod 11 = (2-4)mod11 = 9
(200*301) mod 11 = (2*4)mod11 = 8

$10 \bmod 7 = 3 \rightarrow 10^n \bmod 7 = (10 \bmod 7)^n = 3^n \bmod n$

2.3.5 Inverses

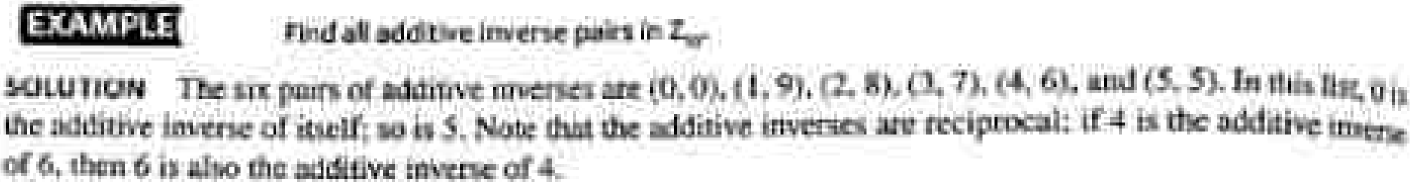
- When we are working in modular arithmetic, we often need to find the inverse of a number relative to an operation.
- We are normally looking for an **additive inverse** or a **multiplicative inverse**.

Additive inverse

- In Z_n, two numbers a and b are additive inverses of each other if **$a + b \equiv 0 \pmod{n}$**
- In Z_n, the additive inverse of a can be calculated as **b = n - a**. For example, the additive inverse of 4 in Z_{10} is $10 - 4 = 6$.

> In modular arithmetic, each integer has an additive inverse. The sum of an integer and its additive inverse is congruent to 0 modulo n.

Note that in modular arithmetic, each number has an additive inverse and the inverse is unique; each number has one and only one additive inverse. However the inverse of the number may be the number itself.



EXAMPLE Find all additive inverse pairs in Z_{10}.

SOLUTION The six pairs of additive inverses are (0, 0), (1, 9), (2, 8), (3, 7), (4, 6), and (5, 5). In this list, 0 is the additive inverse of itself; so is 5. Note that the additive inverses are reciprocal; if 4 is the additive inverse of 6, then 6 is also the additive inverse of 4.

Multiplicative Inverse

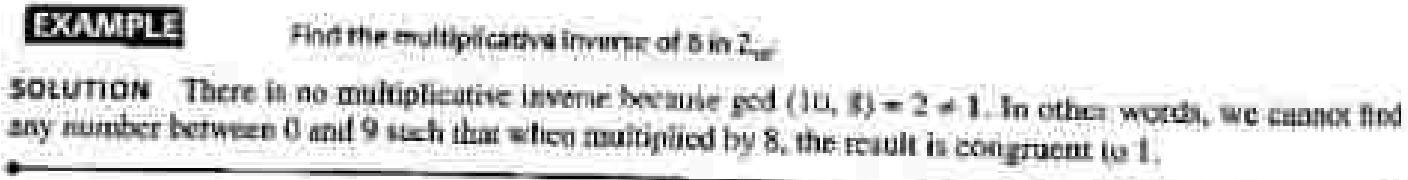
- In Z_n, two numbers a and b are multiplicative inverses of each other if

  $a \times b \equiv 1 \pmod{n}$

- For example, if the modulus is 10, then the multiplicative inverse of 3 is 7. In other words, we have $(3 \times 7) \bmod 10 = 1$.

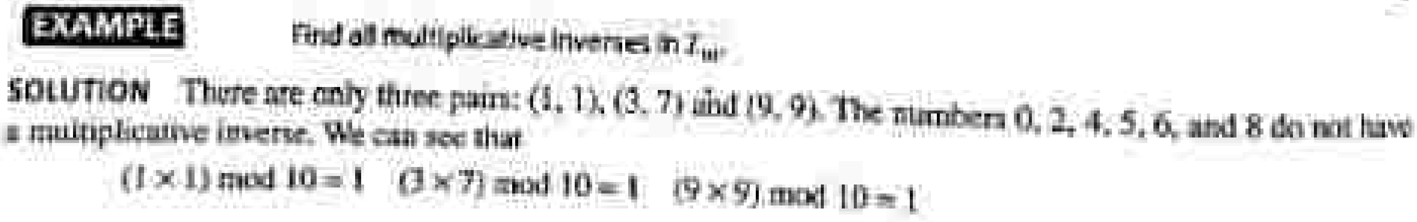
> In modular arithmetic, an integer may or may not have a multiplicative inverse. When it does, the product of the integer and its multiplicative inverse is congruent to 1 modulo n.

It can be proved that a has a multiplicative inverse in Z_n if and only if **gcd (n, a) = 1**. In this case, a and n are said to be **relatively prime**.



EXAMPLE Find the multiplicative inverse of 8 in Z_{10}.

SOLUTION There is no multiplicative inverse because $\gcd(10, 8) = 2 \neq 1$. In other words, we cannot find any number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.



EXAMPLE Find all multiplicative inverses in Z_{10}.

SOLUTION There are only three pairs: (1, 1), (3, 7) and (9, 9). The numbers 0, 2, 4, 5, 6, and 8 do not have a multiplicative inverse. We can see that

$$(1 \times 1) \bmod 10 = 1 \quad (3 \times 7) \bmod 10 = 1 \quad (9 \times 9) \bmod 10 = 1$$

EXAMPLE Find all multiplicative inverse pairs in Z_{11}

SOLUTION We have seven pairs: (1, 1), (2, 6), (3, 4), (5, 9), (7, 8), (9, 5) , and (10, 10). In moving from $\mathbf{Z}_{10}$ to $\mathbf{Z}_{11}$, the number of pairs doubles. The reason is that in $\mathbf{Z}_{11}$, gcd (11, a) is 1 (relatively prime) for all values of a except 0. It means all integers 1 to 10 have multiplicative inverses.

The integer a in Z_n has a multiplicative inverse if and only if gcd (n, a) ≡ 1 (mod n)

The extended Euclidean algorithm we can find the multiplicative inverse of b in Z_n when n and b are given and inverse exists.

To show this, let us replace the first integer a with n (the modulus). We can say that the algorithm can find s and t such $s \times n + b \times t = \gcd(n, b)$.

However, if the multiplicative inverse of b exists, gcd (n, b) must be 1. So the relationship is

$(s \times n) + (b \times t) = 1$

Now we apply the modulo operator to both sides. In other words, we map each side to Z_n. We will have

Now we apply the modulo operator to both sides. In other words, we map each side to $\mathbf{Z}_n$. We will have

$(s \times n + b \times t) \bmod n = 1 \bmod n$

$[(s \times n) \bmod n] + [(b \times t) \bmod n] = 1 \bmod n$

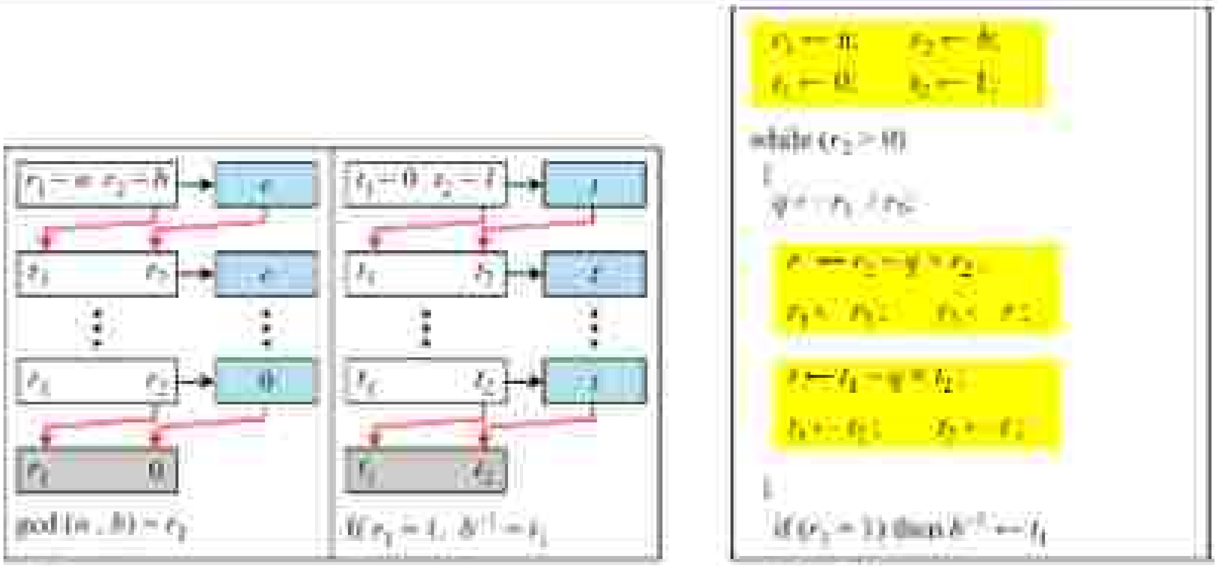
$0 + [(b \times t) \bmod n] = 1$

$(b \times t) \bmod n = 1$ → This means t is the multiplicative inverse of b in $\mathbf{Z}_n$

Note that $[(s \times n) \bmod n]$ in the third line is 0 because if we divide $(s \times n)$ by n, the quotient is s but the remainder is 0.

The extended Euclidean algorithm finds the multiplicative inverses of b in Z_n when n and b are given and gcd (n, b) = 1.

The multiplicative inverse of b is the value of t after being mapped to Z_n.



$r_1 \leftarrow n$ $r_2 \leftarrow b$
q
r_1 r_2
r
r_1 r_2
0
r_1 0
gcd (n , b) = r_1
$t_1 \leftarrow 0$ $t_2 \leftarrow 1$
t
t_1 t_2
t
t_1 t_2
t
t_1 t_2
If $r_1 = 1$, $b^{-1} = t_1$

a. Process

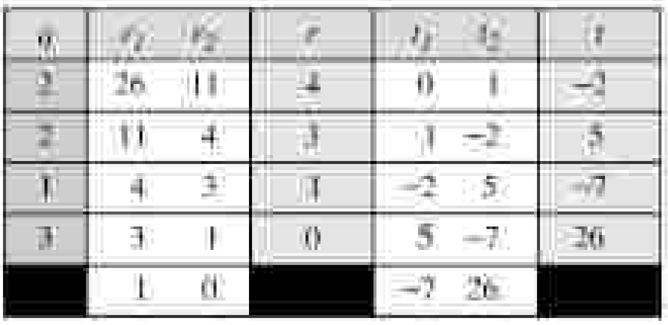
```
$r_1 \leftarrow n;$      $r_2 \leftarrow b;$
$t_1 \leftarrow 0;$      $t_2 \leftarrow 1;$
while ($r_2 > 0$)
{
   $q \leftarrow r_1 / r_2;$

   $r \leftarrow r_1 - q \times r_2;$
   $r_1 \leftarrow r_2;$      $r_2 \leftarrow r;$

   $t \leftarrow t_1 - q \times t_2;$
   $t_1 \leftarrow t_2;$      $t_2 \leftarrow t;$
}
if ($r_1 = 1$) then $b^{-1} \leftarrow t_1$
```

b. Algorithm

Figure 2.9 using the extended Euclidean algorithm to find the multiplicative inverse

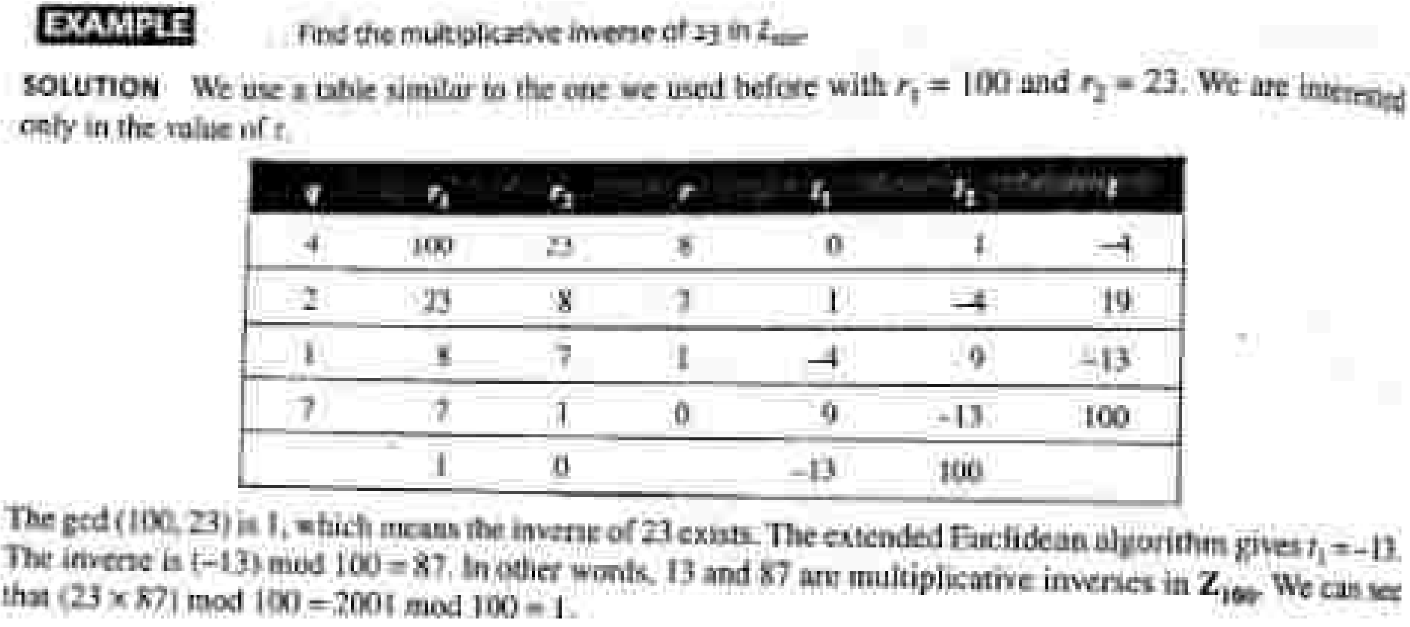
Figure 2.9 shows how we find the multiplicative inverse of a number using the extended Euclidean algorithm.

Example:
Find the multiplicative inverse of 11 in Z_{26}.
We use a table similar to the one we used before with $r_1 = 26$, $r_2 = 11$. We are interested only in the value of t.



q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	−2
2	11	4	3	1	−2	5
1	4	3	1	−2	5	−7
3	3	1	0	5	−7	26
	1	0		−7	26	

The gcd (26, 11) is 1, which means that the multiplicative inverse of 11 exists. The extended Euclidean algorithm gives $t_1 = -7$. The multiplicative inverse is (-7) mod 26 = 19. In other words, 11 and 19 are multiplicative inverse in Z_{26}. We can see that (11 × 19) mod 26 = 209 mod 26 = 1.

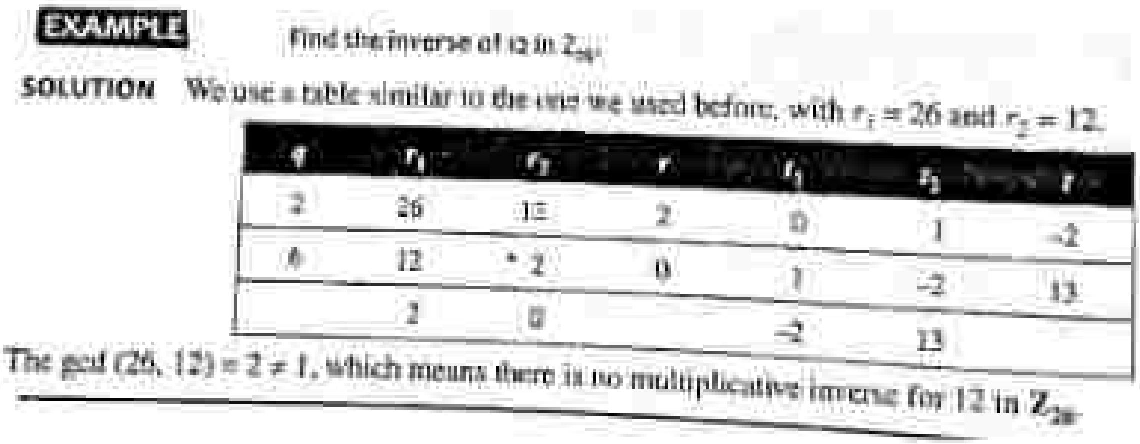


EXAMPLE Find the multiplicative inverse of 23 in Z_{100}.

SOLUTION We use a table similar to the one we used before with $r_1 = 100$ and $r_2 = 23$. We are interested only in the value of t.

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	−4
2	23	8	7	1	−4	19
1	8	7	1	−4	9	−13
7	7	1	0	9	−13	100
	1	0		−13	100	

The gcd (100, 23) is 1, which means the inverse of 23 exists. The extended Euclidean algorithm gives $t_1 = -13$. The inverse is (−13) mod 100 = 87. In other words, 13 and 87 are multiplicative inverses in $\mathbf{Z}_{100}$. We can see that (23 × 87) mod 100 = 2001 mod 100 = 1.



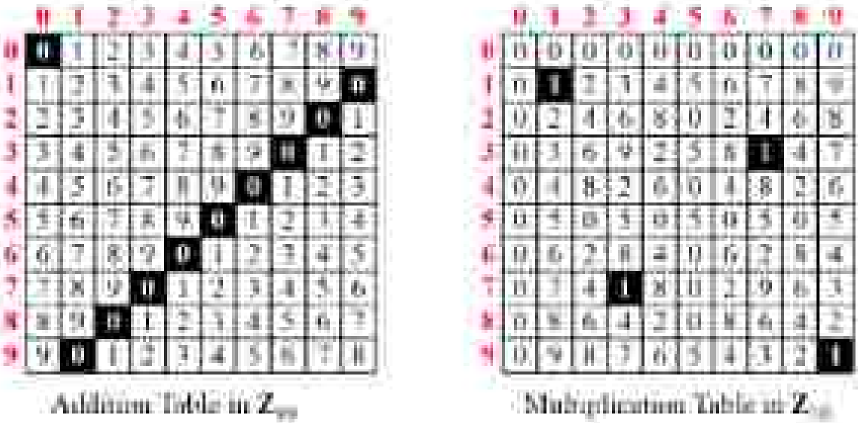
EXAMPLE Find the inverse of 12 in Z_{26}.

SOLUTION We use a table similar to the one we used before, with $r_1 = 26$ and $r_2 = 12$.

q	r_1	r_2	r	t_1	t_2	t
2	26	12	2	0	1	−2
6	12	2	0	1	−2	13
	2	0		−2	13	

The gcd (26, 12) = 2 ≠ 1, which means there is no multiplicative inverse for 12 in $\mathbf{Z}_{26}$.

2.3.6 Addition and Multiplication Tables



	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Addition Table in $\mathbf{Z}_{10}$

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

Multiplication Table in $\mathbf{Z}_{10}$

Figure 2.10 Addition and multiplication tables for Z_{10}

- Figure 2.10 shows two tables for addition and multiplication.
- In addition table, each integer has an additive inverse. ***The inverse pairs can be found when the result of addition is zero.*** **((a + b) mod 10 = 0)**
- In multiplication table, we have only three multiplicative pairs (1, 1), (3, 7), (9, 9). ***The pairs can be found whenever the result of multiplication is 1.*** **((a * b) mod 10 = 1)**

Different Sets for Addition and Multiplication:

We need to use Z_n when additive inverses are needed; we need to use Z_{n^*} when multiplicative inverses are needed.

$\mathbf{Z}_6 = \{0, 1, 2, 3, 4, 5\}$

$\mathbf{Z}_6^* = \{1, 5\}$

$\mathbf{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$

$\mathbf{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$

$\mathbf{Z}_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\mathbf{Z}_{10}^* = \{1, 3, 7, 9\}$

Figure 2.11 Some Z_n and Z_n^* sets

2.3.7 Two more sets

- The set Z_p is same as Z_n except that n is prime. Z_p contains all integers from 0 to p-1.Each member in Z_p has an additive inverse; each member except 0 has a multiplicative inverse.
- The set Z_{p^*} is same as Z_{n^*} except that n is prime. Z_{p^*} contains all integers from 1 to p-1.Each member in Z_{p^*} has an additive and multiplicative inverse.
- The following shows these two sets when p=13

  Z_{13} = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

  Z_{13^*} = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

2.4 Matrices

In cryptography, we need to handle matrices. The following brief review of matrices is necessary preparation for the study of cryptography.

2.4.1 Definition

A matrix is a rectangular array of *l* × *m* elements, in which *l* is the number of rows and *m* is the number of columns.

A matrix is normally denoted with a boldface uppercase letter such as **A**. The element a_{ij} is located in the *i*th row and *j*th column. Although the elements can be a set of numbers.

$$\text{Matrix } \mathbf{A}: \begin{array}{cc} & m \text{ columns} \\ l \text{ rows} & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{l1} & a_{l2} & \cdots & a_{lm} \end{bmatrix} \end{array}$$

Figure 2.12 A matrix of size *l X m*

- If a matrix has only one row (*l = 1*), it is called a row matrix; if it has only one column (*m = 1*) it is called column matrix.
- In a square matrix, in which there is the same number of rows and columns (*l = m*)
- An *additive identity matrix*, denoted as **0**, is a matrix with all rows and columns set to 0's.
- An *identity matrix*, denoted as **1**, is a square matrix with 1s on the main diagonal and 0s elsewhere.

$$\underbrace{\begin{bmatrix} 2 & 1 & 5 & 11 \end{bmatrix}}_{\text{Row matrix}} \quad \underbrace{\begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}}_{\text{Column matrix}} \quad \underbrace{\begin{bmatrix} 23 & 14 & 56 \\ 12 & 21 & 18 \\ 10 & 8 & 31 \end{bmatrix}}_{\text{Square matrix}} \quad \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{0}} \quad \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{I}}$$

Figure 2.13 Examples of matrices

2.4.2 Operations and Relations

In linear algebra, one relation (equality) and four operations (addition, subtraction, multiplication and scalar multiplication) are defined for matrices.

Equality

Two matrices are equal if they have the same number of rows and columns and the corresponding elements are equal. In other words, A = B if we have $a_{ij} = b_{ij}$ for all *i*'s and *j*'s.

Addition and Subtraction

$$\begin{bmatrix} 12 & 4 & 4 \\ 11 & 12 & 30 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} + \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

$$\begin{bmatrix} -2 & 0 & -2 \\ -5 & -8 & -10 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} - \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

$$\mathbf{D} = \mathbf{A} - \mathbf{B}$$

Figure 2.14 Addition and subtraction of matrices

Multiplication

Figure 2.15 shows the product of a row matrix (1 × 3) by a column matrix (3 × 1). The result is a matrix of size 1 × 1.

$$\begin{array}{ccccc} \mathbf{C} & & \mathbf{A} & & \mathbf{B} \\ \begin{bmatrix} 53 \end{bmatrix} & = & \begin{bmatrix} \underrightarrow{5 \quad 2 \quad 1} \end{bmatrix} & \times & \begin{bmatrix} 7 \\ 8 \\ 2 \end{bmatrix}\Big\downarrow \end{array}$$

In which $\boxed{53 = 5 \times 7 + 2 \times 8 + 1 \times 2}$

Figure 2.15 Multiplication of arrow matrix by a column matrix

$$\begin{array}{ccccc} \mathbf{C} & & \mathbf{A} & & \mathbf{B} \\ \begin{bmatrix} 52 & 18 & 14 & 9 \\ 41 & 21 & 22 & 7 \end{bmatrix} & = & \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix} & \times & \begin{bmatrix} 7 & 3 & 2 & 1 \\ 8 & 0 & 0 & 2 \\ 1 & 3 & 4 & 0 \end{bmatrix} \end{array}$$

Figure 2.16 Multiplication of a 2 x 3 matrix by a 3 x 4 matrix

Scalar Multiplication

$$\begin{array}{ccccc} \mathbf{B} & & & & \mathbf{A} \\ \begin{bmatrix} 15 & 6 & 3 \\ 9 & 6 & 12 \end{bmatrix} & = & 3 & \times & \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix} \end{array}$$

Figure 2.17 Scalar multiplication

Determinant

The determinant of a square matrix A of size m x m denoted as det (A) is scalar calculated recursively as shown below:

1. If m=1, det (A)=a_{11}
2. If m>1, det(A) = $\sum(-1)^{i+j}$ x a_{ij} x det(A_{ij})

where A_{ij} is a matrix obtained from A by deleting the i^{th} row and j^{th} column.

Example: we can calculate the determinant of a 2X 2 matrix

$$\det\begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det[4] + (-1)^{1+2} \times 2 \times \det[3] \longrightarrow 5 \times 4 - 2 \times 3 = 14$$

or $$\boxed{\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} \times a_{22} - a_{12} \times a_{21}}$$

Inverses:
Matrices have both additive and multiplicative inverses.

Additive inverse:
The additive inverse of matrix A is another matrix B such that A + B = 0.

In other words, we have $b_{ij} = -a_{ij}$ for all values of i and j. normally the additive inverse of A is defined by −A.

Multiplicative inverse:
The multiplicative inverse is defined only for square matrices. The multiplicative inverse of a square matrix A is a square matrix B such that A x B = B x A = I.

Normally the multiplicative inverse of A is defined by A^{-1}. However, matrices with real elements have inverses only if det (A) ≠ 0.

Multiplicative inverses are only defined for square matrix

Residue Matrices:
Cryptography uses residue matrices: matrices with all elements are in Z_n. All operations on residue matrices are performed the same as for the integer matrices except that the operations are done in modular arithmetic. The residue matrix has a multiplicative inverse if gcd (det (A), n) = 1.

Example

A residue matrix and its multiplicative inverse in Z_{26}

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 7 & 2 \\ 1 & 4 & 7 & 2 \\ 6 & 3 & 9 & 17 \\ 13 & 5 & 4 & 16 \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} 15 & 21 & 0 & 15 \\ 23 & 9 & 0 & 22 \\ 15 & 16 & 18 & 3 \\ 24 & 7 & 15 & 3 \end{bmatrix}$$

$$\det(\mathbf{A}) = 21 \qquad \det(\mathbf{A}^{-1}) = 5$$

Congruence: Two matrices are congruent modulo n, written as A ≡ B (mod n), if they have the same number of rows and columns and all corresponding elements are congruent modulo n. In other words A ≡ B (mod n) if $a_{ij} \equiv b_{ij}$ for all i's and j's.

5. Linear Congruence:

5.1 Single Variable Linear Equations:

Let us see how we can solve equations involving a single variable—that is, equations of the form $ax \equiv b \pmod{n}$. An equation of this type might have no solution or a limited number of solutions. Assume that the gcd $(a, n) = d$. If $d \nmid b$, there is no solution. If $d \mid b$, there are d solutions.

If $d \mid b$, we use the following strategy to find the solutions:

1. Reduce the equation by dividing both sides of the equation (including the modulus) by d.
2. Multiply both sides of the reduced equation by the multiplicative inverse of a to find the particular solution x_0.
3. The general solutions are $x = x_0 + k\,(n / d)$ for $k = 0, 1, \ldots, (d - 1)$.

Example:

Solve the equation 14x ≡ 12 mod 18.

Solution:

The equation is of the form ax ≡ b mod n

Step1:

d = gcd(a, n)

= gcd(14, 18)

= gcd(18, 14)

= gcd(14, 4)

= gcd(4, 2)

= gcd(2 , 0)

= 2

Since 2 | 12, we have 2 solutions

Step2:

Reduce the equation by dividing both sides of the equation by d

$$\frac{14x}{2} \equiv \frac{12}{2} \text{ mod } \frac{18}{2}$$

7x = 6 mod 9

$x_0 = (6 * 7^{-1}) \text{ mod } 9$

now we need to find 7^{-1} with respect to mod 9

(m * 7) mod 9 = 1

(4 * 7) mod 9 = 1 ⇒ inverse of 7 is 4

$\Rightarrow x_0 = (6 * 4) \text{ mod } 9$

$x_0 = 6$

$x_1 = x_0 + 1*(18/2) = 15$

Both solutions, 6 and 15 satisfy the congruence relation, because (14*6) mod 18 = 12 and also (14*15) mod 18 = 12.

Example Solve the equation $3x + 4 \equiv 6 \pmod{13}$.

SOLUTION First, we change the equation to the form $ax \equiv b \pmod{n}$. We add −4 (the additive inverse of 4) to both sides, which give $3x \equiv 2 \pmod{13}$. Because gcd (3, 13) = 1, the equation has only one solution, which is $x_0 = (2 \times 3^{-1}) \bmod 13 = 18 \bmod 13 = 5$. We can see that the answer satisfies the original equation $3 \times 5 + 4 \equiv 6 \pmod{13}$.

Exercise:

Find all solutions to each of the following linear equations:

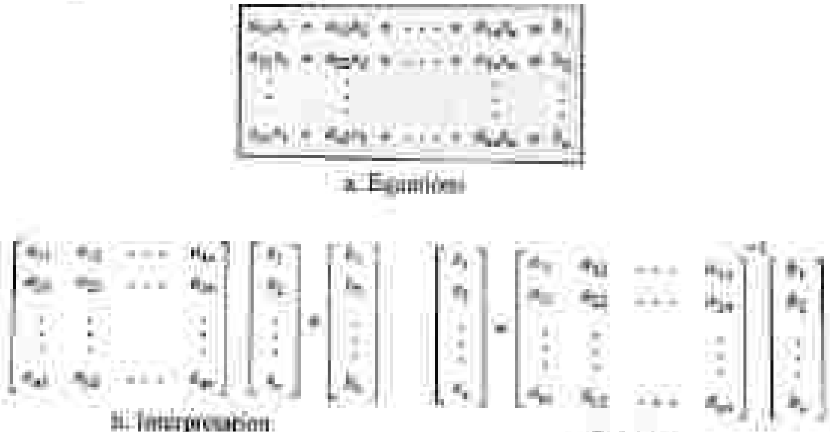
a) 3x ≡ 4 mod 6 b) 4x ≡ 4 mod 6 c) 9x ≡ 12 mod 7

d) 256 ≡ 442 mod 60 e) 3x + 5 ≡ 4 mod 5 f) 4x + 6 ≡ 4 mod 6

g) 9x+4 ≡ 12 mod 7 h) 232x + 42 ≡ 248 mod 50

5.2 Set of linear equations:

- We can also solve a set of linear equations with the same modulus if the matrix formed from the coefficients of the variables is invertible.
- We make three matrices. The first is the square matrix made from the confidents of variables
- The second is a column matrix made from the variables.
- The third is a column matrix made from the values at the right - hand side of the congruence operator.
- We can interpret the set of equations as matrix multiplication.
- If both sides of congruence are multiplied by the multiplicative inverse of the first matrix, the result is the variable matrix at the right hand side, which means the problem can be solved by a matrix multiplication as shown below



a. Equations

b. Interpretation

c. Solution

Example:

Solve the set of following three equations:

2x + 7y + 3z ≡ 4 (mod 16)

$x + 4y + 13z \equiv 5 \pmod{16}$
$3x + 5y + 7z \equiv 3 \pmod{16}$

The matrix form of these equations could be,

$$\begin{bmatrix} 2 & 7 & 3 \\ 1 & 4 & 13 \\ 3 & 5 & 7 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \equiv \begin{bmatrix} 4 \\ 5 \\ 3 \end{bmatrix} \quad \text{mod } 16$$

The solution for these equation is,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 & 7 & 3 \\ 1 & 4 & 13 \\ 3 & 5 & 7 \end{bmatrix}^{-1}\begin{bmatrix} 4 \\ 5 \\ 3 \end{bmatrix} \quad \text{mod } 16$$

The solution to this matrix equations generates the values of x, y, z as follows.

$x \equiv 15(\text{mod } 16)$

$y \equiv 4(\text{mod } 16)$

$z \equiv 14(\text{mod } 16)$.

R16 - CNS - Syllabus topics in prescribed textbook

Prescribed book Details	
Cryptography and Network Security, Behrouz A Forouzan, Debdeep Mukhopadhyay, (3e) Mc Graw Hill.	
Topics covered unit wise	
Unit-I	**Ch-1 Ch-2** **1.1** **1.2** **1.3**
Unit-II	**Ch-4** **Ch-5** **Ch-6** **Ch-7**
Unit-III	**Ch-9** **Ch-10**
Unit-IV	**Ch-11** **Ch-12** **Ch-13** **Ch-15**
Unit-V	**Ch-16** **Ch-17** **Ch-18** **Ch-19**