Chapter9:
Network Management and Security:

Introduction to Network Management:

Network management is defined as monitoring, testing, configuring, and troubleshooting network components to meet a set of requirements defined by an organization. These requirements include the smooth, efficient operation of the network that provides the predefined quality of service for users. To accomplish this task, a network management system uses hardware, software, and humans.

Functions of Network Management System:

1. Configuration Management
2. Fault Management
3. Performance Management
4. Security management
5. Accounting management

Configuration Management

A large network is usually made up of hundreds of entities that are physically or logically connected to one another. These entities have an initial configuration when the network is set up, but can change with time. Desktop computers may be replaced by others; application software may be updated to a newer version; and users may move from one group to another. The configuration management system must know, at any time, the status of each entity and its relation to other entities. Configuration management can be subdivided into two parts reconfiguration and Documentation.

Fault Management:

Falls on two categories.

- Reactive Fault Management
  A reactive fault management system is responsible for detecting, isolating, correcting, and recording faults. It handles short-term solutions to faults.
- Proactive Fault Management
  Proactive fault management tries to prevent faults from occurring. Although this is not always possible, some types of failures can be predicted and prevented.

Performance management:

It is is closely related to fault management and tries to monitor and control the network to ensure that it is running as efficiently as possible.

Security Management

Security management is responsible for controlling access to the network based on the predefined policy.

Accounting Management

Accounting management is the control of users' access to network resources through charges. Charging does not necessarily mean cash transfer; it may mean debiting the departments or divisions for budgeting purposes. Today, organizations use an accounting management system for the following reasons:

- It prevents users from monopolizing limited network resources.
- It prevents users from using the system inefficiently.
- Network managers can do short- and long-term planning based on the demand for network use.

Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is an "Internet-standard protocol for managing devices on IP networks. Devices that typically support SNMP include routers, switches, servers, workstations, printers, modem racks, and more. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention.

The Simple Network Management Protocol (SNMP) is a framework for managing devices in an Internet using the TCPIIP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an Internet.

Concept

SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers . SNMP is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks.

Managers and Agents

A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent. The agent keeps performance information in a database. The manager has access to the values in the database. For example, a router can store in appropriate variables the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not.

An SNMP-managed network consists of three key components:

- Managed device
- Agent — software which runs on managed devices
- Network management system (NMS) — software which runs on the manager

A managed device is a network node that implements an SNMP interface that allows unidirectional (read-only) or bidirectional access to node-specific information. Managed devices exchange node-specific information with the NMSs. Sometimes called network elements, the managed devices can be any type of device, including, but not limited to, routers, access servers, switches, bridges, hubs, IP telephones, IP video cameras, computer hosts, and printers.

An agent is a network-management software module that resides on a managed device. An agent has local knowledge of management information and translates that information to or from an SNMP specific form.

A network management system (NMS) executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs may exist on any managed network.

Management with SNMP is based on three basic ideas:

- A manager checks an agent by requesting information that reflects the behavior of the agent.
- A manager forces an agent to perform a task by resetting values in the agent database.
- An agent contributes to the management process by warning the manager of an unusual situation.

SNMP operates in the Application Layer of the Internet Protocol Suite (Layer 7 of the OSI model). The SNMP agent receives requests on UDP port 161. The manager may send requests from any available source port to port 161 in the agent. The agent response will be sent back to the source port on the manager. The manager receives notifications (Traps and InformRequests) on port 162. The agent may generate notifications from any available port.

To do management tasks, SNMP uses two other protocols:

- Structure of Management Information (SMI)
- Management Information Base (MIB).

Role of SNMP

SNMP has some very specific roles in network management. It defines the format of the packet to be sent from a manager to an agent and vice versa. It also interprets the result and creates statistics (often with the help of other management software). The packets exchanged contain the object (variable) names and their status (values). SNMP is responsible for reading and changing these values.

Roles of SMI

SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values. SM1 does not define the number of objects an entity should manage or name the objects to be managed or define the association between the objects and their values.

The Structure of Management Information, version 2 (SMIv2) is a component for network management. Its functions are

- To name objects
- To define the type of data that can be stored in an object
- To show how to encode data for transmission over the network

SMI is a guideline for SNMP. It emphasizes three attributes to handle an object: name, data type, and encoding method .

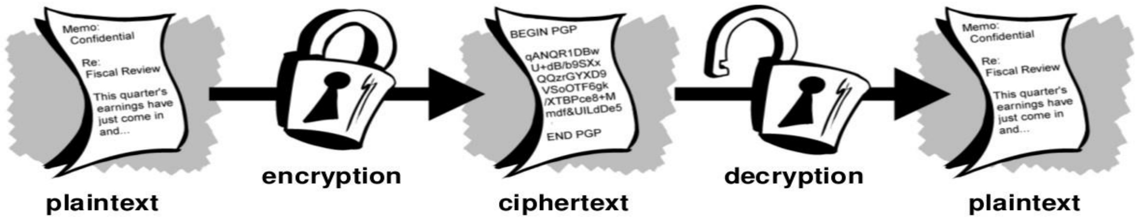
Roles of MIB

For each entity to be managed, this protocol must define the number of objects, name them according to the rules defined by SMI, and associate a type to each named object .MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed. Each agent has its own MIB2, which is a collection of all the objects that the manager can manage. The objects in MIB2 are categorized under 10 different groups: system, interface, address translation, ip, icmp, tcp, udp, egp, transmission, and snmp.

Cryptography

Cryptography is derived from the Greek words: kryptós, "hidden", and gráphein, "to write" - or "hidden writing". Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

Encryption and Decryption



Plain-text and Cipher-text

The original message, before being transformed, is called plaintext. After the message is transformed, it is called cipher-text. An encryption algorithm transforms the plain text into cipher text; a decryption algorithm transforms the cipher-text back into plain- text. The sender uses an encryption algorithm, and the receiver uses a decryption algorithm.

Cipher

We refer to encryption and decryption algorithms as ciphers. The term cipher is also used to refer to different categories of algorithms in cryptography. This is not to say that every sender-receiver pair needs their very own unique cipher for a secure communication. On the contrary, one cipher can serve millions of communicating pairs.

Key

A key is a number (or a set of numbers) that the cipher, as an algorithm, operates on. To encrypt a message, we need an encryption algorithm, an encryption key, and the plain-text. These create the cipher-text. To decrypt a message, we need a decryption algorithm, a decryption key, and the cipher-text. These reveal the original plain-text.

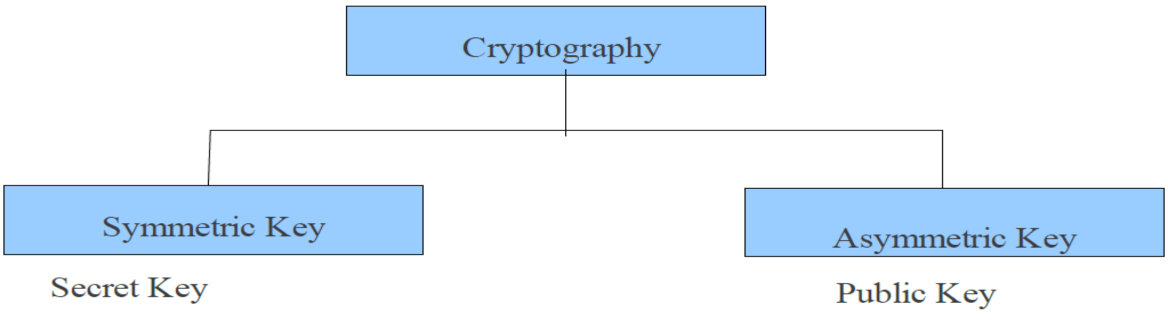
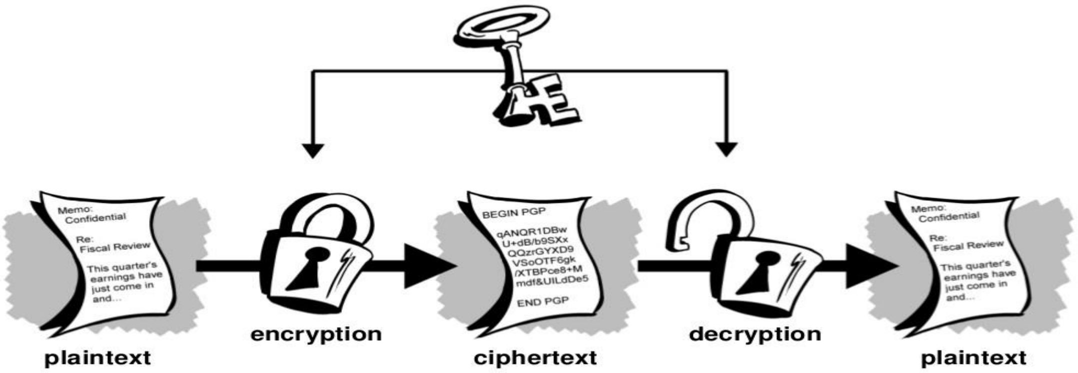


Fig:Categories of Cryptography

Symmetric-key

In conventional cryptography, also called secret-key or symmetric-key encryption, one key is used both for encryption and decryption. The Data Encryption Standard (DES) is an example of a conventional cryptosystem that is widely employed by the Federal Government. Figure below shows an illustration of the conventional encryption process.

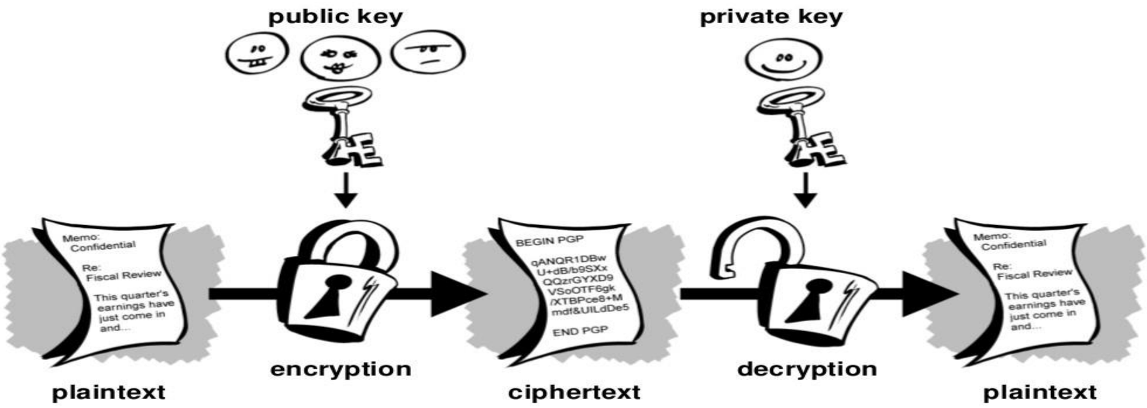

Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext
encryption
BEGIN PGP
qANQR1DBw U+dB/b9SXx QQzrGYXD9 VSoOTF6gk /XTBPce8+M mdf&UILdDe5
END PGP
ciphertext
decryption
Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext

Conventional encryption has benefits. It is very fast. It is especially useful for encrypting data that is not going anywhere. However, conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution.

For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key.

Asymmetric-Key Cryptography

Public key cryptography is an asymmetric scheme that uses a pair of keys for encryption: a public key, which encrypts data, and a corresponding private, or secret key for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read . It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.


public key
private key
Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext
encryption
BEGIN PGP
qANQR1DBw U+dB/b9SXx QQzrGYXD9 VSoOTF6gk /XTBPce8+M mdf&UILdDe5
END PGP
ciphertext
decryption
Memo: Confidential
Re: Fiscal Review
This quarter's earnings have just come in and...
plaintext

The Essential steps in Asymmetric-key cryptography are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the keys in a public register or other accessible file. This is the public key. The companion key is kept private. Each user maintains a collection of public keys obtained from others
3. If Bob wishes to send a private message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows the Alice's private key.

With this approach, all the participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user protects his and her private key, incoming communication is secure. At any time, a user change the private key and publish the companion public key replace the old public key.

Comparison

Let us compare symmetric-key and asymmetric-key cryptography. Encryption can be thought of as electronic locking; decryption as electronic unlocking. The sender puts the message in a box and locks the box by using a key; the receiver unlocks the box with a key and takes out the message. The difference lies in the mechanism of the locking and unlocking and the type of keys used. In symmetric-key cryptography, the same key locks and unlocks the box. In asymmetric-key cryptography, one key locks the box, but another key is needed to unlock it.

Traditional Cipher used in Symmetric-key Cryptography:
Two types:

1. Substitution cipher
2. Transposition cipher

Substitution cipher:

A substitution cipher substitutes one symbol with another. If the symbols in the plain- text are alphabetic characters, we replace one character with another. For example, we can replace character A with D, and character T with Z. If the symbols are digits (0 to 9), we can replace 3 with 7, and 2 with 6. It is also known and Ceaser's Cipher who invented it.

For example, if we encode the word “SECRET” using Caesar’s key value of 3, we offset the alphabet so that the 3rd letter down (D) begins the alphabet. So starting with ABCDEFGHIJKLMNOPQRSTUVWXYZ and sliding everything up by 3, you get DEFGHIJKLMNOPQRSTUVWXYZABC where D=A, E=B, F=C, and so on. Using this scheme, the plaintext, “SECRET” encrypts as “VHFUHW.” To allow someone else to read the cipher text, you tell them that the key is 3.

Transposition Ciphers

In a transposition cipher, there is no substitution of characters; instead, their locations change. A character in the first position of the plaintext may appear in the tenth position of the cipher text. A character in the eighth position may appear in the first position. In other words, a transposition cipher reorders the symbols in a block of symbols.

Key: In a transposition cipher, the key is a mapping between the position of the symbols in the plaintext and cipher text. For example, the following shows the key using a block of four characters:

Plaintext: 2 4 1 3
Cipher text: 1 2 3 4
In encryption, we move the character at position 2 to position 1, the character at position 4 to position 2, and so on. In decryption, we do the reverse.

Encryption algorithm:
The most commonly used symmetric encryption are block ciphers. A block cipher processes the plain text input in fixed size blocks and produces a block of cipher text of equal size for each plain text block.
The two most important symmetric algorithms, both of which are block ciphers, are
Data Encryption Standard (DES)
Advanced Encryption Standard (AES)

DES (Data Encryption Standard):
The Data Encryption Standard, is a block cipher operating on 64-bit data blocks. DES was designed by IBM and adopted by the U.S. government as the standard encryption method for nonmilitary and nonclassified use. The algorithm encrypts a 64-bit plaintext block using a 64-bit key, as shown in Figure

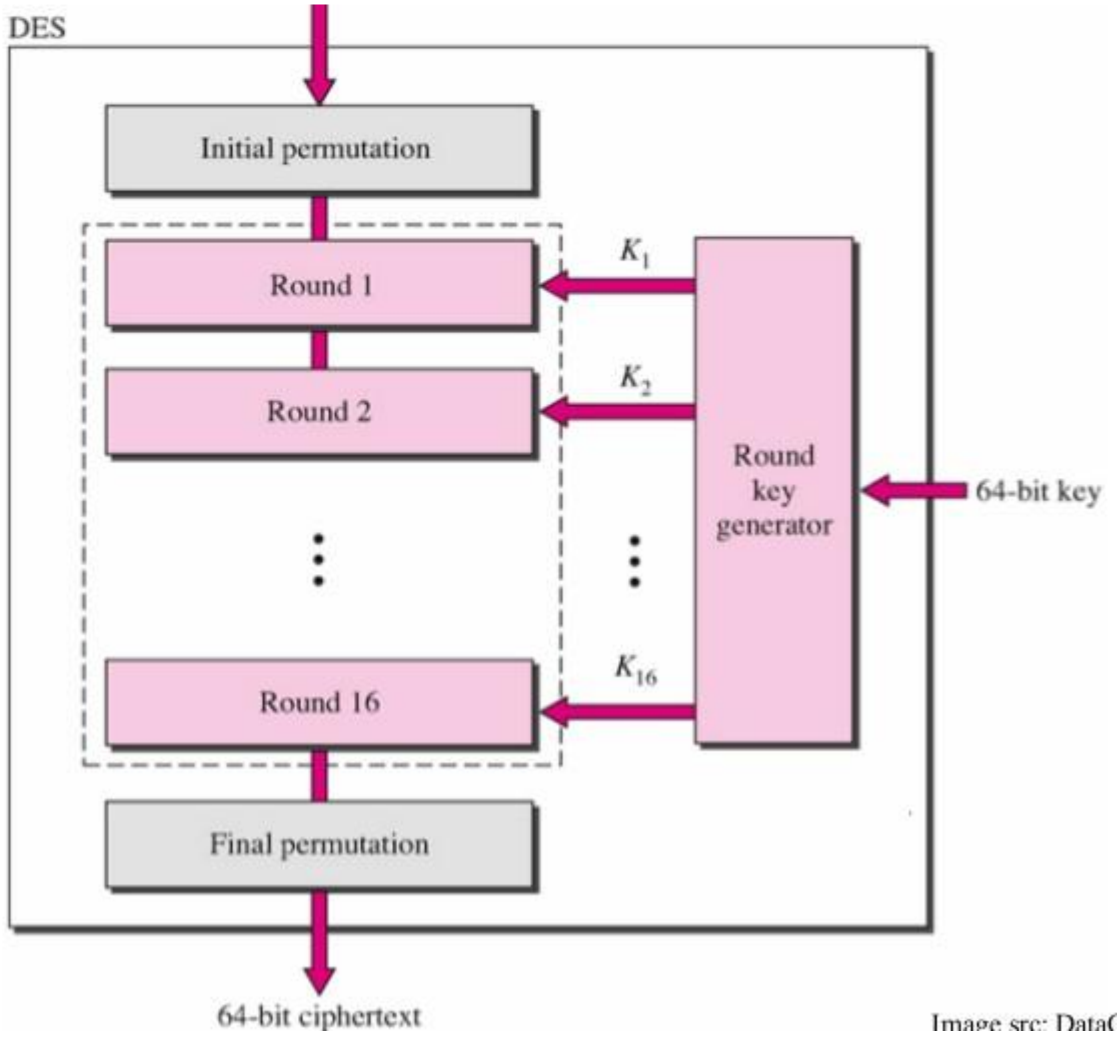
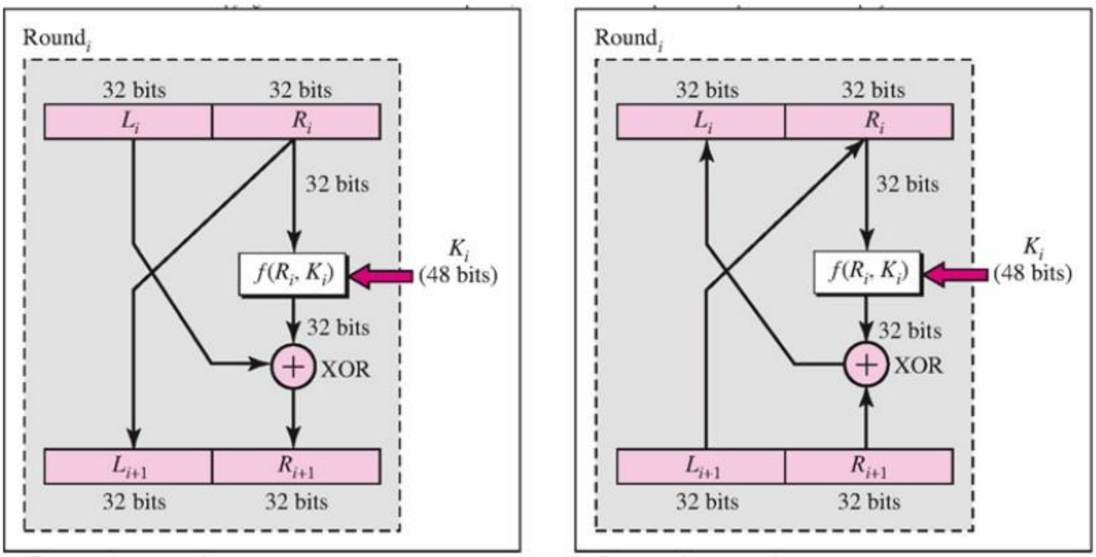


Image src: Data(

DES has two transposition blocks (P-boxes) and 16 complex round ciphers (they are repeated). Although the 16 iteration round ciphers are conceptually the same, each uses a different key derived from the original key. The initial and final permutations are keyless straight permutations that are the inverse of each other. The permutation takes a 64-bit input and permutes them according to predefined values. Each round of DES is a complex round cipher, as shown in Figure below. Note that the structure of the encryption round ciphers is different from that of the decryption one.



a. Encryption round b. Decryption round

Asymmetric Key Cryptography:
Some examples of public-key cryptosystems are :
Elgamal (named for its inventor, Taher Elgamal),
RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman),
Diffie-Hellman (named for its inventors),
DSA ,the Digital Signature Algorithm (invented by David Kravitz).

RSA Algorithm

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- **Generate the RSA modulus (n)**
    - Select two large primes, p and q.

  - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

- **Find Derived Number (e)**
  - Number **e** must be greater than 1 and less than (p − 1)(q − 1).
  - There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.

- **Form the public key**
  - The pair of numbers (n, e) form the RSA public key and is made public.
  - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

- **Generate the private key**
  - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
  - Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).
  - This relationship is written mathematically as follows −

$$ed = 1 \bmod (p - 1)(q - 1)$$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and (p − 1)(q − 1) = 6 × 12 = 72, except for 1.
- The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.
- Check that the d calculated is correct by computing −

$de = 29 \times 5 = 145 = 1 \bmod 72$

- Hence, public key is (91, 5) and private keys is (91, 29).

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e).
- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$C = P^e \bmod n$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$C = 10^5 \bmod 91$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$\text{Plaintext} = C^d \bmod n$

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

$\text{Plaintext} = 82^{29} \bmod 91 = 10$

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- **Encryption Function** − It is considered as a one-way function of converting plaintext into cipher text and it can be reversed only with the knowledge of private key d.
- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.