Chapter2
Basic of Modern cryptography

One Way Function

A trapdoor function' is a function that is easy to compute in one direction, yet believed to be difficult to compute in the opposite direction (finding its inverse) without special information, called the "trapdoor". Trapdoor functions are widely used in cryptography.

An example of a simple mathematical trapdoor is "6895601 is the product of two prime numbers. What are those numbers?"
- A typical solution would be to try dividing 6895601 by several prime numbers until finding the answer. However, if one is told that 1931 is part of the answer, one can find the answer by entering "6895601 ÷ 1931" into any calculator.

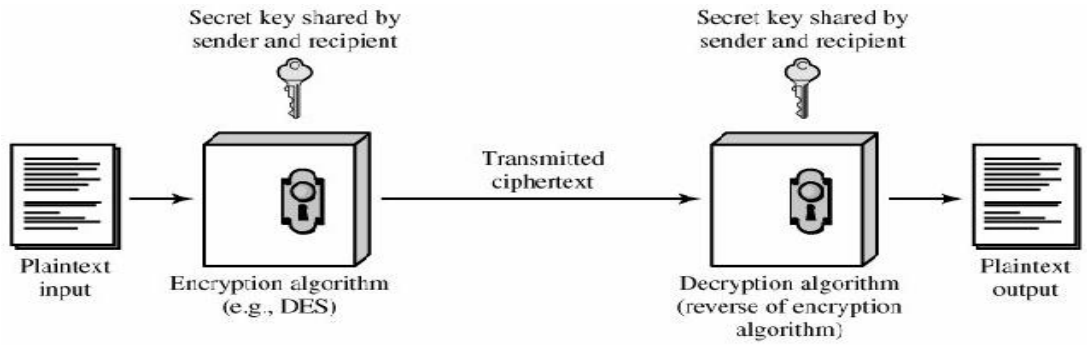
This example is not a sturdy trapdoor function--modern computers can guess all of the possible answers within a second--but this sample problem could be improved by using the product of two much larger primes.

Symmetric and Asymmetric cryptography

Symmetric Cipher Model

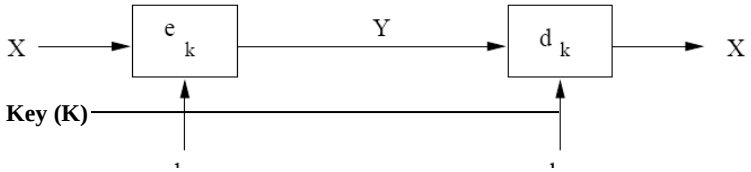
A symmetric encryption scheme has five ingredients as shown in figure:

1. Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.
2. Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.
3. Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. 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.
4. 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. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
5. Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.



Asymmetric Cipher model

Review of Symmetric Cryptography



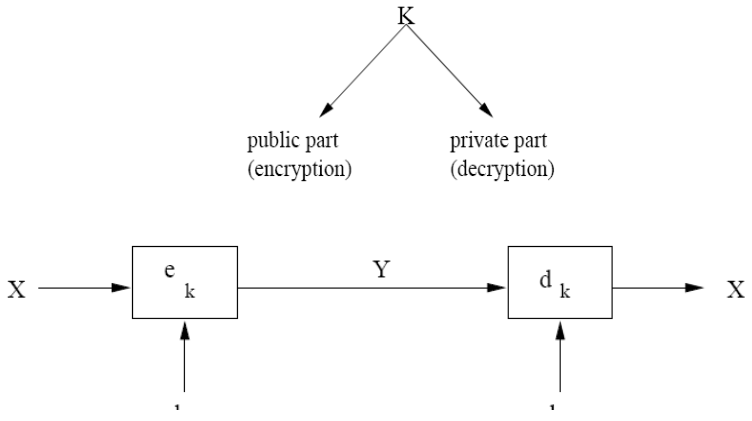
Two properties of symmetric-key schemes:
1. The algorithm requires same secret key for encryption and decryption.
2. Encryption and decryption are essentially identical (symmetric algorithms).

Analogy for symmetric key algorithms: Symmetric key schemes are like a safe box with a strong lock. Everyone with the key can deposit and retrieve messages.

Main problems with symmetric key schemes are:
1. Requires secure transmission of secret key.
2. In a network environment, each pair of users has to have a different key resulting in too many keys (n (n -1)/ 2 key pairs.

New Idea: Make a slot in the safe box so that everyone can deposit a message, but only the receiver can open the safe and look at the content of it. Idea: **Split key**.

Public Key Cryptography Protocol (Model)



Encryption Key (K_E) **Decryption Key (K_D)**

1. Alice and Bob agree on a public-key cryptosystem.
2. Bob sends Alice his public key.
3. Alice encrypts her message with Bob's public key and sends the ciphertext.
4. Bob decrypts ciphertext using his private key.

Mechanisms that can be realized with public-key algorithms

1. Key establishment protocols (e.g., Diffie-Hellman key exchange) and key transport protocols (e.g., via RSA) without prior exchange of a joint secret.
2. Digital signature algorithms (e.g., RSA, DSA)
3. Encryption

There are three families of Public-Key (PK) algorithms of practical relevance:

1. Integer factorization algorithms (RSA, ...)
2. Discrete logarithms (Diffie-Hellman, DSA, ...)
3. Elliptic curves (EC)

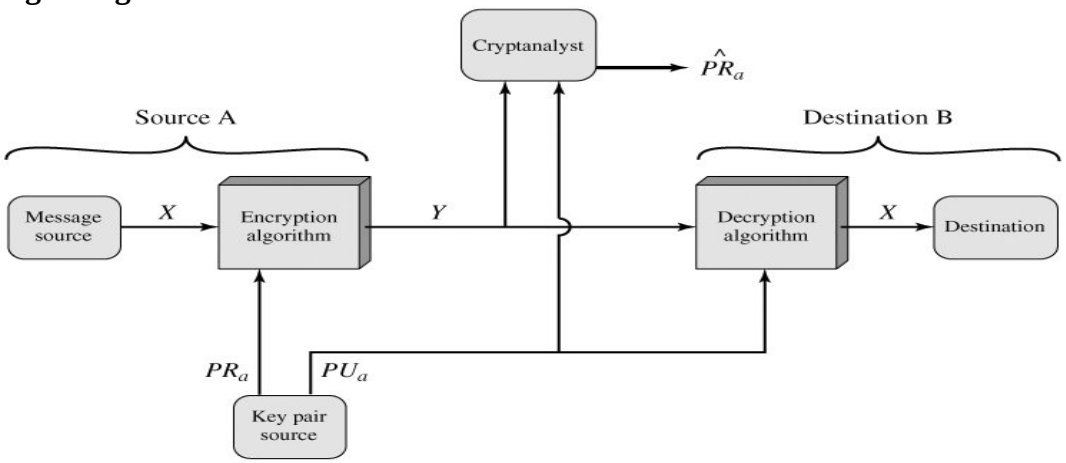
In this lecture we only consider Algorithms of family i.e. Integer Factorization Algorithm and Discrete Logarithms.

Comparison between Symmetric and asymmetric

	Secret Key (Symmetric)	**Public Key (Asymmetric)**
Number of keys	1	2
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse; secrecy and integrity data—single characters to blocks of data, messages, files	Key exchange, authentication
Key	Must be out-of-band	Public key can be used to distribute

	Secret Key (Symmetric)	Public Key (Asymmetric)
distribution		other keys
Speed	Fast	Slow; typically, 10,000 times slower than secret key

Digital signature



In this case, A prepares a message to B and encrypts it using A's private key before transmitting it. B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message. Therefore, the entire encrypted message serves as a **digital signature**. In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.

In the preceding scheme, the entire message is encrypted, which, although validating both author and contents, requires a great deal of storage. Each document must be kept in plaintext to be used for practical purposes

A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such a block, called an **authenticato**r, must have the property that it is infeasible to change the document without changing the authenticator.

If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing. This is called **Hash Authentication** and we discuss it later in detail.