**Cryptographic**

Inhoud

[Chapter 1: Overview 2](#_Toc376001663)

[Cryptographic algorithms and protocols can be grouped into four main areas 2](#_Toc376001664)

[Field of network and internet security 2](#_Toc376001665)

[Computer Security Objectives 2](#_Toc376001666)

[Possible Additional concepts 3](#_Toc376001667)

[Breach of Security: Levels of Impact 3](#_Toc376001668)

[Computer Security Challenges 3](#_Toc376001669)

[OSI Security Architecture 3](#_Toc376001670)

[Threats and attacks 4](#_Toc376001671)

[Security attacks 4](#_Toc376001672)

[Passive attack: more in depth 4](#_Toc376001673)

[Active attack: more in depth 4](#_Toc376001674)

[Security Services 5](#_Toc376001675)

[X.800 Service Categories 5](#_Toc376001676)

[Model for Network Security 6](#_Toc376001677)

[Network Access Security Model 7](#_Toc376001678)

[Unwanted Access 7](#_Toc376001679)

[Chapter 2: Classical Encryption Techniques 7](#_Toc376001680)

[Symmetric Encryption 7](#_Toc376001681)

[Basic Terminology 7](#_Toc376001682)

[Simplified Model of Symmetric Encryption 8](#_Toc376001683)

[Model of Symmetric Cryptosystem. 8](#_Toc376001684)

[Cryptographic Systems 9](#_Toc376001685)

[Cryptanalysis and Brute-Force Attack 9](#_Toc376001686)

[Encryption Scheme Security 10](#_Toc376001687)

[Brute-Force Attack 10](#_Toc376001688)

[Substitution Technique 11](#_Toc376001689)

[Monoalphabetic Cipher 11](#_Toc376001690)

[Playfair Cipher 11](#_Toc376001691)

[Hill Cipher 12](#_Toc376001692)

[Polyalphabetic Ciphers 12](#_Toc376001693)

[Vigenère Cipher 12](#_Toc376001694)

[Vigenère Autokey System 13](#_Toc376001695)

[Vernam Cipher. 13](#_Toc376001696)

[One – Time Pad 13](#_Toc376001697)

[Rail Fence Cipher 14](#_Toc376001698)

[Row Transposition Cipher 14](#_Toc376001699)

[Steganography 14](#_Toc376001700)

[Part 3: Block Ciphers and the Data Encryption Standard. 15](#_Toc376001701)

[Stream Cipher 15](#_Toc376001702)

[Block Cipher 15](#_Toc376001703)

[Feistel Cipher 15](#_Toc376001704)

[Diffusion and Confusion 16](#_Toc376001705)

[Feistel Cipher Design Features 16](#_Toc376001706)

[Feistel Example 17](#_Toc376001707)

[Data Encryption Standard (DES) 17](#_Toc376001708)

[Strength of DES 18](#_Toc376001709)

[Block Cipher Design Principles: Number of rounds. 18](#_Toc376001710)

[Block Cipher Design Principles: Design of Function F 18](#_Toc376001711)

[Block Cipher Design Principles: Key Schedule Algorithm 18](#_Toc376001712)

[Chapter 4: Basic Concepts in Number Theory and Finite Fields 19](#_Toc376001713)

[Divisibility 19](#_Toc376001714)

[Properties of divisibility 19](#_Toc376001715)

[Division Algorithm 19](#_Toc376001716)

[Euclidean Algorithm 19](#_Toc376001717)

[Greatest Common Divisor (GCD) 19](#_Toc376001718)

[Modular Arithmetic 20](#_Toc376001719)

[Properties of Congruences 20](#_Toc376001720)

[Modular Arithmetic: properties 20](#_Toc376001721)

[Groups 21](#_Toc376001722)

[Cyclic Group 21](#_Toc376001723)

[Rings 21](#_Toc376001724)

[Fields 22](#_Toc376001725)

[Finite Fields of the form GF (p) 23](#_Toc376001726)

[Polynomial Arithmetic 23](#_Toc376001727)

[Polynomial Arithmetic With Coefficients in Zp 23](#_Toc376001728)

[Polynomial Division 24](#_Toc376001729)

[Polynomial GCD 24](#_Toc376001730)

[Computational Considerations 24](#_Toc376001731)

[Using a Generator 24](#_Toc376001732)

[Chapter 5: Advanced Encryption Standard 25](#_Toc376001733)

[Finite Field Arithmetic 25](#_Toc376001734)

[S-Box Rationale 25](#_Toc376001735)

[Shift Row Rationale 26](#_Toc376001736)

[Mix Columns Rationale 26](#_Toc376001737)

[AddRoundKey Transformation 26](#_Toc376001738)

[AES Key Expansion 26](#_Toc376001739)

[Key Expansion Rationale 27](#_Toc376001740)

[Equivalent Inverse Cipher 27](#_Toc376001741)

[Interchanging: InvShiftRows and InvSubBytes. 27](#_Toc376001742)

[Interchanging: AddRoundKey and InvMixColumns 27](#_Toc376001743)

[Implementation Aspects 28](#_Toc376001744)

[Chapter 9: Public Key Cryptography and RSA 28](#_Toc376001745)

[Misconceptions concerning Public-Key Encryption 28](#_Toc376001746)

[Principles of Public Key Cryptosystems 28](#_Toc376001747)

[Public Key Cryptosystems 29](#_Toc376001748)

[Public Key Cryptosystem: Secrecy 30](#_Toc376001749)

[Public Key Cryptosystem: Authentication 30](#_Toc376001750)

[Public Key Cryptosystem: Authentication and Secrecy 30](#_Toc376001751)

[Applications for Public-Key Cryptosystems 31](#_Toc376001752)

[Public Key Requirements 32](#_Toc376001753)

[Public Key Cryptanalysis 32](#_Toc376001754)

[Rivest-Shamir-Adleman Scheme (RSA) 33](#_Toc376001755)

[RSA Algorithm 33](#_Toc376001756)

[Algorithm Requirements 33](#_Toc376001757)

[Example of RSA Algorithm 34](#_Toc376001758)

[Exponentiation in Modular Arithmetic 34](#_Toc376001759)

[Efficient Operation Using the Public Key 34](#_Toc376001760)

[Efficient Operation Using the Private Key 34](#_Toc376001761)

[Key Generation 34](#_Toc376001762)

[The Security of RSA 35](#_Toc376001763)

[Factoring Problem 35](#_Toc376001764)

[Timing Attacks 35](#_Toc376001765)

[Countermeasures 36](#_Toc376001766)

[Fault-Based attack 36](#_Toc376001767)

[Chosen Ciphertext attack (CCA) 36](#_Toc376001768)

[Chapter 10: Other Public-Key Cryptosystems 37](#_Toc376001769)

[Diffie-Hellman Key Exchange 37](#_Toc376001770)

[Key Exchange Protocols 37](#_Toc376001771)

[ElGamal Cryptography 38](#_Toc376001772)

[Elliptic Curve Arithmetic 38](#_Toc376001773)

[Elliptic Curves over Zp 38](#_Toc376001774)

[Elliptic Curves over GF (2m) 38](#_Toc376001775)

[Elliptic Curve Cryptography (ECC) 39](#_Toc376001776)

[ECC Encryption / Decryption 39](#_Toc376001777)

[Security of Elliptic Curve Cryptography 39](#_Toc376001778)

# Chapter 1: Overview

## Cryptographic algorithms and protocols can be grouped into four main areas

* Symmetric encryption
  + = used to conceal the contents or blocks or streams of data of any size, including messages, files, encryption keys and passwords
* Asymmetric encryption
  + used to conceal small blocks of data, such as encryption keys and hash function values,, which are used in digital signatures
* data integrity algorithms
  + used to protect blocks of data, such as messages, from alteration
* Authentication protocols
  + schemes based on the use of cryptographic algorithms designed to authenticate the identity of entities

## Field of network and internet security

consists of measures to

* defer
* prevent
* detect
* correct

security violations that involve the transmission of information.

## Computer Security Objectives

* Confidentiality
  + data confidentiality
    - Assures that private or confidential information is not made available or disclosed to unauthorized individuals
  + Privacy
    - Assures that individuals control or influence what information related to them may be collected and stored and by 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, Integrity, Availability )

## Possible Additional concepts

* Authenticity
  + Verifying that users are who they say they are and that each input arriving at the system came from a trusted source
* Accountability
  + The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
  + Supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after action recovery and legal action.
  + Tracing a security breach to a responsible party

## Breach of Security: Levels of Impact

* Loss
  + the loss could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals
* Moderate
  + The loss could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals.
* High
  + The loss could be expected to have **a severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals

## Computer Security Challenges

* not simple
* potential attacks on the security features need to be considered
* procedures used to provide particular services are often counter-intuitive
* it is necessary to decide where to use the various security mechanisms
* requires constant monitoring
* is too often an afterthought
* security mechanisms typically involve more than a particular algorithm or protocol
* security is essentially a battle of wits between a perpetrator and the designer
* little benefit from security investment is perceived until a security failure occurs
* strong security is often viewed as an impediment to efficient and user-friendly operation

## OSI Security Architecture

* Security Attack
  + Any action that compromises the security of information owned by the organization.
* Security Mechanism
  + A process (or a device incorporating such a process) that is designed to detect, prevent or recover form 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
  + Intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

## Threats and attacks

Threat

= A potential for violation of security, which exists when there is a circumstance, capability, action or event that could breach the security and cause harm.

* possible danger that might exploit a vulnerability

Attack

= An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

## Security attacks

* passive attack
  + attempts to learn or make use of the information of the system but does not affect system resources
* active attack
  + attempts to alter system resources or affect their operation

## Passive attack: more in depth

* are in the nature of eavesdropping on , or monitoring of, transmissions
* Goal? obtain information that is being transmitted.
* Two types:
  + the release of message contents
  + traffic analysis

## Active attack: more in depth

* involve some modification of the data stream or the creation of a false stream
* difficult to prevent because of the wide variety of potential physical, software and network vulnerabilities
* Goal is to detect attacks and to recover from any disruption or delays caused by them.

Forms:

* Masquerade
  + Takes place when one entity pretends to be a different entity
  + usually includes one of the other forms of active attack
* Replay
  + Involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect
* Modification of messages
  + some portion of a legitimate message is altered, or messages are delayed or reordered to produce an unauthorized effect
* Denial of Service (DoS)
  + Prevents or inhibits the normal use or management of communications facilities.

## Security Services

* Defined by X.800 as:
  + a service provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

## X.800 Service Categories

* Authentication

Concerned with assuring that a communication is authentic

* + in the case of a single message, assures the recipient that the message is from the source that it claims to be.
  + In the case of on-going interaction, assures the two entities are authentic and that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties.
    - Two specific authentication services defined in X.800
      * Peer entity authentication
      * Data origin authentication
* Access Control

The ability to limit and control the access to host systems and applications via communication 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.

* Data confidentiality

The protection of transmitted data from passive attacks

* + Broadest service protects all user data transmitted between two users over a period of time
  + Narrower forms of service includes the protection of a single message or even specific fields within a message.

The protection of traffic flow from analysis

* + This requires that an attacker is not able to observe the source and destination, frequency, length or other characteristics of the traffic on a communications facility
* Data Integrity

Can apply to a stream of messages, a single message or selected fields within a message.

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.

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

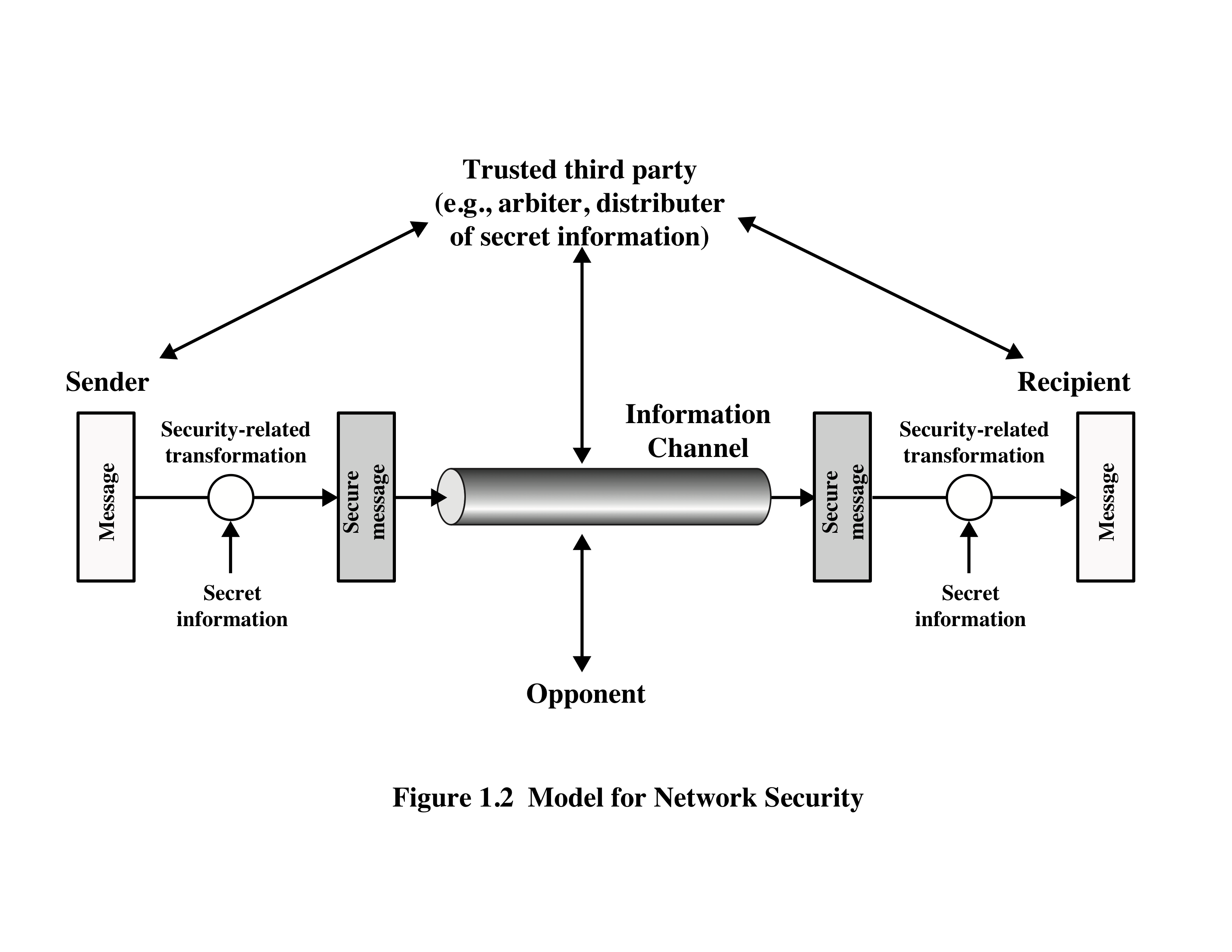
* Nonrepudiation

Prevents either sender or receiver from denying a transmitted message.

When a message is sent, the receiver can prove that the alleged sender in fact sent the message.

When a message is received, the sender can prove that the alleged receiver in fact received the message.

## Model for Network Security



## Network Access Security Model



## Unwanted Access

Placement in a computer system of logic that exploits vulnerabilities in the system and that can effect application programs as well as utility programs such as editors and compilers.

Programs can present two kinds of threats:

* information access threats
  + intercept or modify data on behalf of users who should not have access to that data
* Service threats
  + Exploit service flaws in computers to inhibit use by legitimate users.

# Chapter 2: Classical Encryption Techniques

## Symmetric Encryption

Also referred to as conventional encryption or single-key encryption

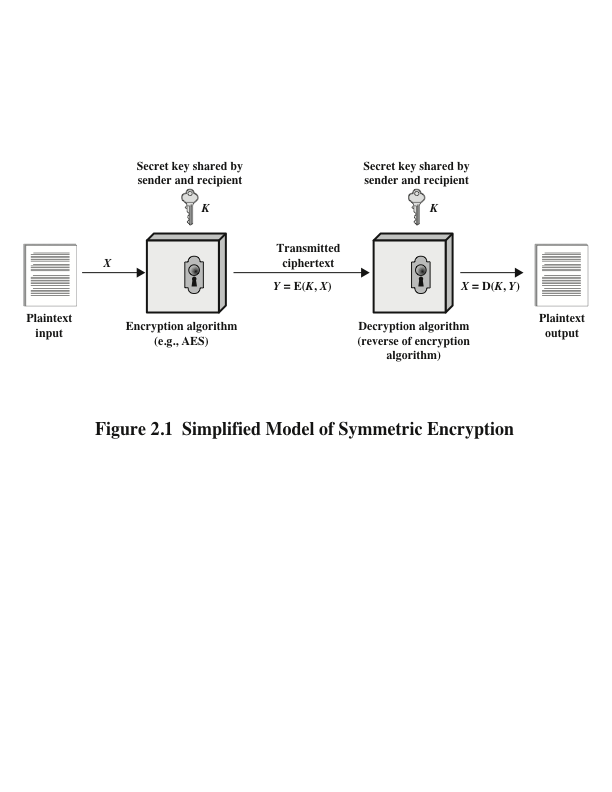
Was the only type of encryption in use prior to the development of public-key encryption in the 1970s.

Remains by far the most widely used of the two types of encryption.

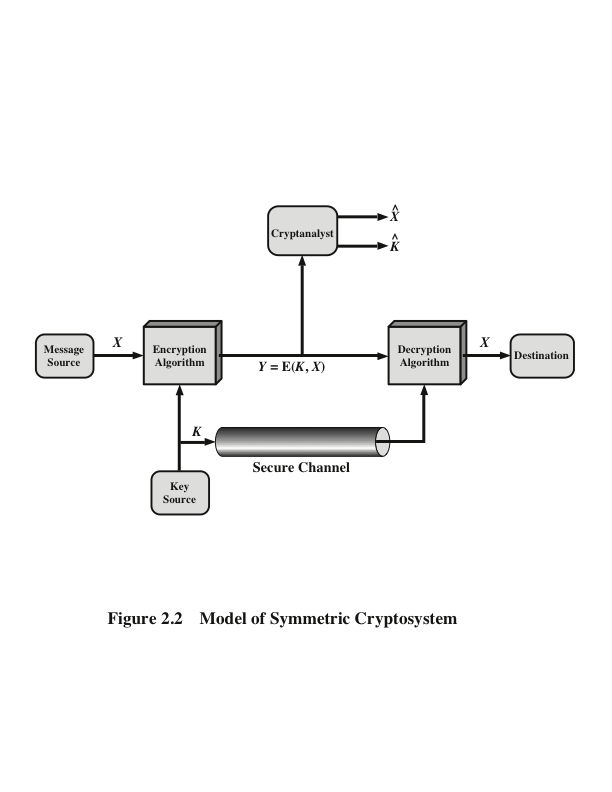
## Basic Terminology

|  |  |
| --- | --- |
| Plaintext | The original message |
| Ciphertext | The coded message |
| Enciphering or encryption | Process of converting from plaintext to ciphertext |
| Deciphering or decryption | Restoring the plaintext from the ciphertext |
| Cryptography | Study of encryption |
| Cryptographic system or cipher | Schemes used for encryption |
| Cryptanalysis | Techniques used for deciphering a message without any knowledge of the enciphering details. |
| Cryptology | Areas of cryptography and cryptanalysis together. |

## Simplified Model of Symmetric Encryption



## Model of Symmetric Cryptosystem.



## Cryptographic Systems

Characterized along three independent dimensions:

* The type of operations used for transforming plaintext to ciphertext
  + Substitution
  + Transportation
* The number of keys used
  + Symmetric, single-key, secret key, conventional encryption
  + Asymmetric, two-key, or public-key encryption
* The way in which the plaintext is processed
  + Block cipher
  + Stream cipher

## Cryptanalysis and Brute-Force Attack

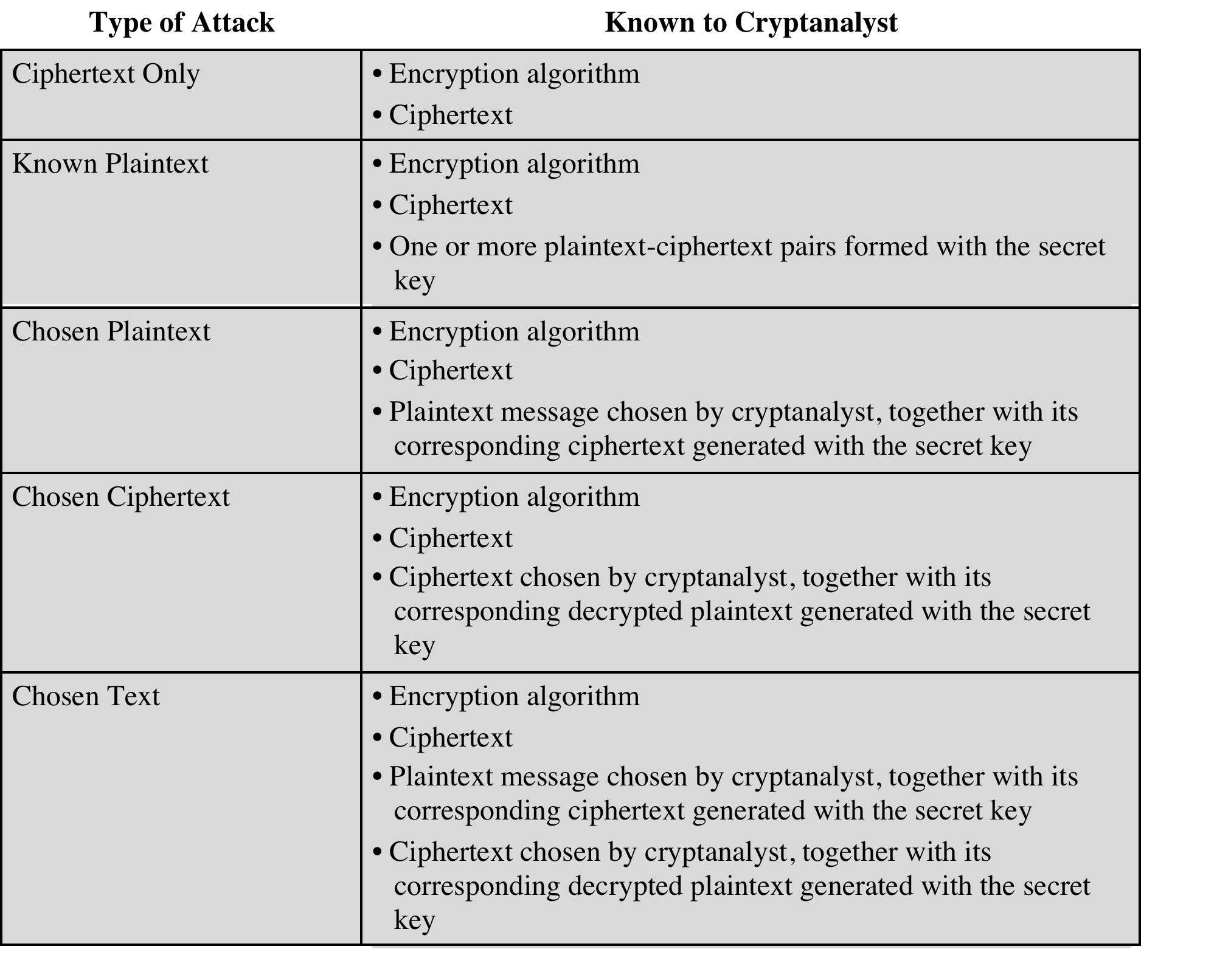
Cryptanalysis:

* Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext.
* Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

Brute-force attack

* Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.
* On average, half of all possible keys must be tried to achieve success.

If either type of attack succeeds tin deducing the key , the effect is catastrophic: All future and past messages encrypted with that key are compromised.



## Encryption Scheme Security

Unconditionally secure

* No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there.

Computationally secure

* The cost of breaking the cipher exceeds the value of the encrypted information.
* This time required to break the cipher exceeds the useful lifetime of the information.

## Brute-Force Attack

* Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.
  + On average, half of all possible keys must be tried to achieve success
    - To supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed.

## Substitution Technique

* is one 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.

## Monoalphabetic Cipher

Permutation

* of a finite se of element S is an ordered sequence of all the element of S, with each element appearing exactly once.

If the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 x 1026 possible keys

* this is 10 orders of magnitude greater than the key space for DES.
* Approach is referred to as a Monoalphabetic substitution cipher because a single cipher alphabet is used per message.

Easy to break because they reflect the frequency data of the original alphabet.

Countermeasure is to provide multiple substitutes (homophones) for a single letter.

Digram

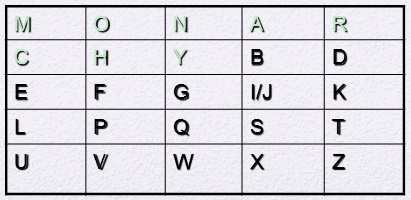
* Two-letter combination
* Most common is “th”

Trigram

* three letter combination
* most frequent is “the“

## Playfair Cipher

* Best-known multiple-letter encryption cipher
* Treats digrams in the plaintext as single units and translates these units into ciphertext digrams.
* Based on the use of a 5x5 matrix of letters constructed using a keyword
* Invented by British scientist Sir Charles Wheatstone in 1854
* Used as the standard field system by the British Army in World War I and the U.S. Army and other Allied forces during World War II.
* Fill in letters of keywords (minus duplicates) from left to right, and from top to bottom, then fill in the remainder of the matrix with the remaining letters in alphabetic order
* using the keyword MONARCHY:



## Hill Cipher

* Developed by the mathematician Lester Hill in 1929
* Strength is that it completely hides single letter frequencies
  + The use of a larger matrix hides more frequency information
  + A 3x3 Hill Cipher hides not only single letter but also two-letter frequency information.
* Strong against a ciphertext –only attack but easily broking with a known plaintext attack.

## Polyalphabetic Ciphers

Polyalphabetic substitution cipher

* improves on the simple Monoalphabetic technique by using different Monoalphabetic substitutions as one proceeds through the plaintext message.

All these techniques have the following features in common:

* A set of related Monoalphabetic substitution rules is used
* A key determines which particular rule is chosen for a given transformation

## Vigenère Cipher

* Best known and one of the simplest polyalphabetic substitution ciphers
* 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.

## Vigenère Autokey System

A keyword is concatenated with the plaintext itself to provide a running key.

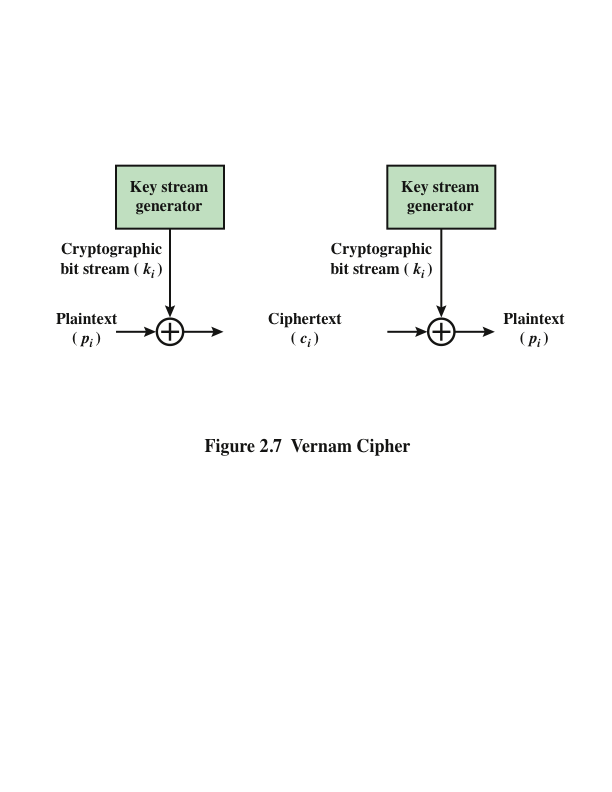
Example:

* key: deceptivewearediscoveredsav
* plaintext: wearediscoveredsaveyourself
* ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

Even this scheme is vulnerable to cryptanalysis.

* because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied.

## Vernam Cipher.



## One – Time Pad

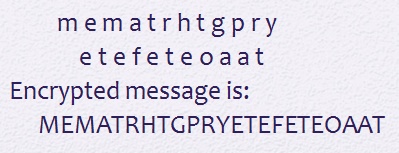
* Improvement to Vernam Cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
* Use a random key that is as long as the message so that they key needs to be repeated.
* Key is used to encrypt and decrypt a single message and then is discarded.
* Each new message requires a new key of the same length as the new message
* Scheme is unbreakable
  + Produces random output that bears no statistical relationship to the plaintext
  + Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code.

Difficulties

* The one-time pad offers complete security but, in practice, has two fundamental difficulties:
  + There is the practical problem of making large quantities of random keys
    - Any heavily used system might require millions of random characters on a regular basis
  + Mammoth key distribution problem
    - For every message to be sent, a key of equal length is needed by both sender and receiver.
* Because of these difficulties, the one-time pad is of limited utility
  + Useful primarily for low-bandwidth channels requiring very high security
* The one-time pad is the only cryptosystem that exhibits perfect secrecy.

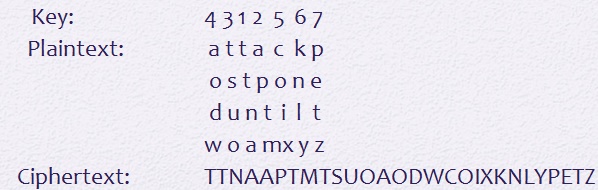
## Rail Fence Cipher

* Simplest transposition cipher
* Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.
* To encipher the message “meet me after the toga party” with a rail fence of depth 2, we would write:



## Row Transposition Cipher

* Is a more complex transposition
* 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 of the algorithm



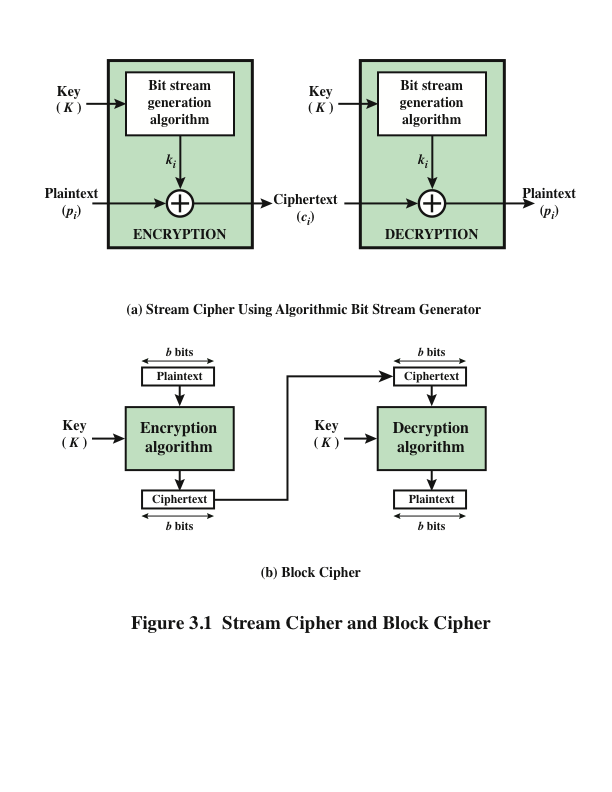
## Steganography

Techniques:

* Character marking
  + Selected letters of printed or typewritten text are over-written in pencil
  + The marks are ordinarily not visible unless the paper is held at an angle to bright light
* Invisible ink
  + A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied
* Pin punctures
  + Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light
* Typewriter correction ribbon
  + Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

# Part 3: Block Ciphers and the Data Encryption Standard.

## Stream Cipher

* Encrypts a digital data stream one bit or one byte at a time
  + examples:
    - Autokeyed Vignenère cipher
    - Vernam Cipher
* In the ideal case a one-time pad version of the Vernam Cipher would be used, in which the keystream is as long as the plaintext bit stream
  + if the cryptographic keystream is random, then the cipher is unbreakable by any means other than acquiring the keystream
    - Keystream must be provided to both users in advance in some independent and secure channel
    - This introduces insurmountable logistical problems if the intended data traffic is very large.
* For practical reasons, the bit-stream generator must be implemented as an algorithmic procedure so that the cryptographic bit stream can be produced by both users
  + It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream.
  + The two users need only share the generating key and each produce the keystream.

## Block Cipher

* A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length
* Typically a block size of 64 or 128 bit is used
* As with a stream cipher, the two users share a symmetric encryption key
* The majority of network-based symmetric cryptographic applications make use of block ciphers.

## Feistel Cipher

* Proposed the use of a cipher that alternates substitutions and permutations
  + Substitution = each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements
  + Permutation = no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.
* Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions
* Is the structure used by many significant symmetric block ciphers currently in use.

## Diffusion and Confusion

Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system

* Shannon’s concern was to thwart cryptanalysis based on statistical analysis.

Diffusion

* The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
* This is achieved by having each plaintext digit affect the value of many ciphertext digits.

Confusion

* seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible.
* Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key.

## Feistel Cipher Design Features

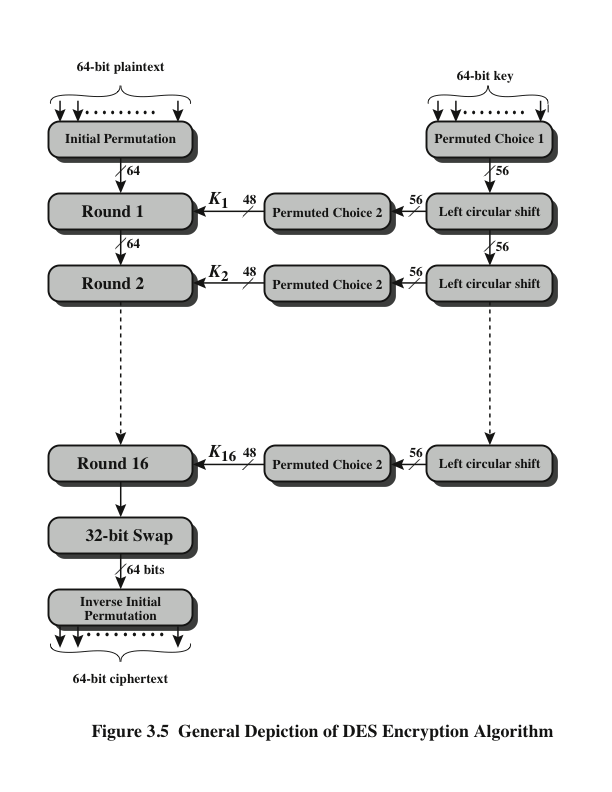
* Block size
  + Larger block sizes mean greater security but reduced encryption / decryption speed for a given algorithm
* Key size
  + Larger key size means greater security but may decrease encryption / decryption speeds.
* Number of rounds
  + The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
* Subkey generation algorithm
  + Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
* Round function F
  + Greater complexity generally means greater resistance to cryptanalysis
* Fast software Encryption / decryption
  + In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
* Ease of analysis
  + If the algorithm can be concisely and clearly explained, it is easier to analyse that algorithm for cryptanalytic vulnerabilities and therefor develop a higher level of assurance as to its strength.

## Feistel Example



## Data Encryption Standard (DES)

* Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46.
* Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
* Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  + Data are encrypted in 64bit blocks using a 56-bit key
  + The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
  + the same steps, with the same key, are used to reserve the encryption



## Strength of DES

Timing Attacks

* one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts.
* Exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.
* So far it appears unlikely that this technique will ever be successful against DES or more powerful symmetric ciphers such as triple DES and AES.

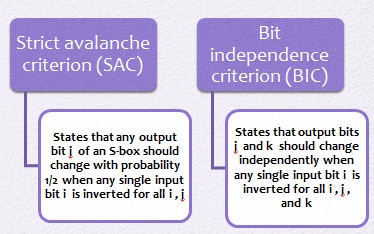
## Block Cipher Design Principles: Number of rounds.

* The greater the number of rounds, the more difficult it is to perform cryptanalysis.
* In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
* if DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search.

## Block Cipher Design Principles: Design of Function F

* The hearth of a Feistel block cipher is the function F.
* The more nonlinear F, the more difficult any type of cryptanalysis will be.
* The SAC and BIC criteria appear to be strengthen the effectiveness of the confusion function

The algorithm should have good avalanche properties.



## Block Cipher Design Principles: Key Schedule Algorithm

* with any Feistel block cipher, the key is used to generate one Subkey for each round.
* In general, we would like to select subkeys to maximize the difficulty of working back to the main key.
* It is suggested that, at a minimum, the key schedule should guarantee key / ciphertext Strict Avalanche Criterion and Bit Independence Criterion.

# Chapter 4: Basic Concepts in Number Theory and Finite Fields

## Divisibility

We say that a nonzero b divides a if a = mb for some m, where a, b and m are integers.

b divides a if there is no remainder on division.

The notations b | a is commonly used to mean b divides a.

If b | a we say that b is a divisor of a.

## Properties of divisibility

* if a | 1, then a = +- 1
* if a | b and b | a, then a = +- B
* Any b =/ 0 divides 0
* If a | b and b | c then a | c
* if b | g and b | h, then b | (mg + nh) for arbitrary integers m and n.
* To see this last point, note that:
  + if b | g, then g is of the form g = b \* g1 for some integer g1
  + if b | h, then is of the form h = b \* h1
* So:

*b* = 7;  *g* = 14; *h* = 63; *m* = 3; *n* = 2

7 | 14 and 7 | 63.

To show 7 (3 \* 14 + 2 \* 63),

we have (3 \* 14 + 2 \* 63) = 7(3 \* 2 + 2 \* 9),

and it is obvious that 7 | (7(3 \* 2 + 2 \* 9)).

* + mg + nh = mbg1 + nbh1 = b \* (mg1 +nh1)

and therefor b divides mg + nh

## Division Algorithm

Given any positive integer n and any nonnegative integer a, if we divide a by n we get an integer quotient q and an integer remainder r that obey the following relationship:

a = qn + r and 0 <= r < n; q = [a/n]

## Euclidean Algorithm

* One of the basic techniques of number theory
* 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 (GCD)

= the greatest common divisor of a and b is the largest integer that divides both a and b.

We can use the notation gcd(a,b) to mean the greatest common divisor of a and b.

We also define that gcd(0,0) = 0.

Positive integer c is said to be the gcd of a and b if:

* c is a divisor of a and b
* any divisor of a and b is a divisor of c
* gcd(a,b) = max[k, such that k | a and k | b]

Because we require that the greatest common divisor be positive

* gcd(a,b) = gcd ( |a|, |b|)

Also because all nonzero integers divide 0, we have gcd (a,0) = | a |.

We stated that 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

## Modular Arithmetic

* The modulus
  + if a is an integer and n is 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:
    - a = qn + r
    - a = [a/n] \* n + ( a mod n)
* Congruent modulo n
  + two integers a and b are said to be congruent module n if (a mod n) = (b mod n)
  + this is written as a = b (mod n)²
  + note that if a = 0 (mod n) then n | a

## Properties of Congruences

* Congruences have the following properties:

1. a = b (mod n) if n (a –b )
2. a = b (mod n) implies b = a (mod n)
3. a = b (mod n) and b = c (mod n) imply a = c (mod n)

* To demonstrate the first point, if n (a –b) then (a – b ) = kn for some k
  + so we can write a = b + kn
  + therefore, (a mod n) = (remainder when b + kn is divided by n) = (remainder when b is divided by n) = (b mod

## Modular Arithmetic: properties

* Modular arithmetic exhibits the following properties:

1. [(*a* mod *n*) + (*b* mod *n*)] mod *n* *= (a + b)* mod *n*
2. [(*a* mod *n*) - (*b* mod *n*)] mod *n = (a - b)* mod *n*
3. [(*a* mod *n*) \* (*b* mod *n*)] mod *n = (a \* b)* mod *n*

* We demonstrate the first property:
  + define ( a mod n) = ra and (b mod n) = rb. Then we can write a = ra + jn for some integer j and b = rb + kn for some integer k
  + Then:

(a + b) mod n = (ra + jn + rb + kn) mod n

= (ra + rb + (k +j)n) mod n

= (ra + rb) mod n

= [(a mod n) + (b mod n)] mod n

## Groups

A set of elements with a binary operation denoted by  that associates that each ordered pair (a, b) of elements in G an element (a  b in G, such that the following axioms are obeyed:

* Closure
  + if a and b belong to G, then a b is also in G
* Associative
  + a (b c) = ( a b) c for all a, b, c in G
* Identity element
  + There is an element e in G such that a e = e a for all a in G
* Inverse element
  + For each a in G , there is an element a in G such that a  a = a  a = e
* Commutative
  + a b = b a for all a, b in G

## Cyclic Group

Exponentiation is defined within a group as a repeated application of the group operator, so that a3 = a a  a .

We define a0 = e as the identity element, and a –n = (a’)n , where a’ is the inverse element of a within the group.

A group G is cyclic if every element of G is a power ak (k is an integer) of a fixed element.

The element a is said to generate the group G or to be a generator of G.

A cyclic group is always abelian and may be finite or infinite.

## Rings

=is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed:

* Closure
* Associative
* Identity element
* Inverse element
* Commutative

Additional axioms:

* Closure under multiplication
* Associativity of multiplication
* Distributive laws

In essence: a ring is a set where we can do addition, subtraction and multiplication without leaving the set.

A ring is said to be commutative if it satisfies the following additional condition:

* Commutativity of multiplication:  
  ab = ba for all a, b in R

An integral domain is a commutative ring that obeys the following axioms:

* Multiplicative identity

There is an element 1 in R such that a 1 = 1a = a for all a in R

* No zero divisors  
  if a, b in R and ab = 0 then either a = 0 or b = 0

## Fields

A field F = set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed:

* Closure
* Associative
* Identity element
* Inverse element
* Commutative
* Closure under multiplication
* Associativity of multiplication
* Distributive laws
* Commutativity of multiplication
* Multiplicative identity
* No zero divisors

And additional:

* Multiplicative inverse:

For each a in F, except 0, there is an element a-1 in F such that aa-1 = (a-1)a = 1

In essence, a field is a set in which we can do addition, subtraction, multiplication and division without leaving the set. Division defined by the following rule: a / b = a (b-1)

Examples: rational numbers, real numbers and complex numbers.

## Finite Fields of the form GF (p)

* Finite fields play a crucial role in many cryptographic algorithms
* it can be shown that the order of an finite field must be a power of a prime pn, where n is a positive integer
  + the only positive integers that are divisor of p are p and 1
* the finite field of order pn is generally written GF(pn)
  + GF stands for Galios Field, in honour of the mathematician who first studied finite fields.

## Polynomial Arithmetic

Three classes:

* Ordinary polynomial arithmetic, using the basic rules of algebra
* Polynomial arithmetic in which the arithmetic on the coefficients is performed module p; that is , the coefficients are in GF (p).
* Polynomial arithmetic in which the coefficients are in GF(p), and the polynomials are defined modulo a polynomial m (x) whose highest power is some integer n.

Example:

let *f(x)* = *x3 + x2* + 2 and *g(x)* = *x2 - x* + 1, where S is the set of integers

Then:

*f(x) + g(x) = x3 + 2x2 - x + 3*

*f(x) - g(x) = x3 + x + 1*

*f(x) \* g(x) = x5 + 3x2 - 2x + 2*

## Polynomial Arithmetic With Coefficients in Zp

* if each distinct polynomial is considered to be an element of the set, then that set is a ring.
* When polynomial arithmetic is performed on polynomials over a field, then division is possible.
  + note: this does not mean that exact division is possible
* If we attempt to perform polynomial division over a coefficient set that is not a field, we find that division is not always defined
  + even if the coefficient set is a field, polynomial division is not necessarily exact
  + with the understanding that remainders are allowed, we can say that polynomial division is possible if the coefficient set is a field.

## Polynomial Division

We can write any polynomial in the form:

*f(x) = q(x) g(x) + r(x)*

* r(x) can be interpreted as being a remainder
* so r(x) = f(x) mod g(x)

If there is no remainder we can say g(x) divides f(x)

A polynomial f(x) over a field F is called irreducible if and only if f(x) cannot be expressed as a product of two polynomials, both over F and both of degree lower than that of f(x).

* An irreducible polynomial is also called a prime polynomial.

## Polynomial GCD

The polynomial c(x) is said to be the greatest common divisor of a(x) and b(x) if the following are true:

* c(x) divides both a(x) and b(x-)
* any divisor of a(x) and b(x) is a divisor of c(x)

The Euclidean algorithm can be extended to find the greatest common divisor of two polynomials whose coefficients are elements of a field.

## Computational Considerations

* Since coefficients are 0 or 1, they can represent any such polynomial as a bit string
* Addition becomes XOR of these bit strings
* Multiplication is SHIFT and XOR
* Modulo reduction is done by repeatedly substituting the highest power with remainder of irreducible polynomial (also SHIFT and XOR)

## Using a Generator

A generator g of finite field F of order q (contains q elements) is an element whose first q -1 powers generate all the nonzero elements of F.

Consider a field F defined by polynomial fx

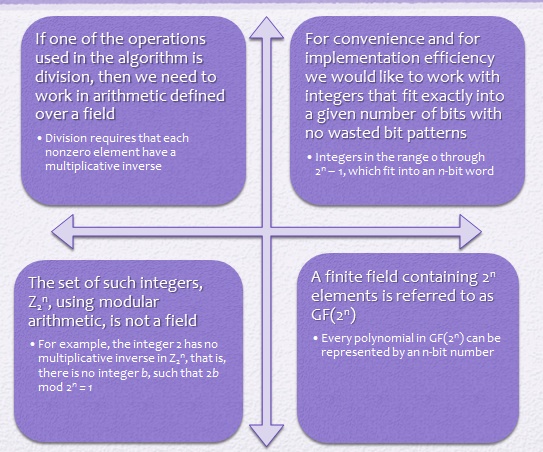
* an element b contained in F is called a root of the polynomial if f(b) = 0

Finally, it can be shown that a root g of an irreducible polynomial is a generator of the finite group defined on that polynomial.

# Chapter 5: Advanced Encryption Standard

## Finite Field Arithmetic

* In the Advanced Encryption Standard (AES) all operations are performed on 8-bit bytes.
* the arithmetic operations of addition, multiplication and division are performed over the finite field GF(28)
* A field is a set in which we can do addition, subtraction, multiplication and division without leaving the set.
* Division is defined with the following rule:
  + a / b = a (b-1)
* An example of a finite field (one with a finite number of elements) is the set Zp consisting off all integers { 0,1,…,p -1}, where p is a prime number and in which arithmetic is carried out modulo p.



## S-Box Rationale

* The S-box is designed to be resistant to know cryptanalytic attacks
* The Rijndael developers sought a design that has a low correlation between input bits and output bits and the property that the output is not a linear mathematical function of the input.
* The nonlinearity is due to the use of the multiplicative inverse.

## Shift Row Rationale

* More substantial than it may first appear
* The State, as well as the cipher input and output is treated as an array of four 4 byte columns
* On encryption, the first 4 bytes of the plaintext are copied to the first column of State and so on.
* The round key is applied to State column by column
  + Thus, a row shift moves an individual byte from one column to another, which is linear distance of a multiple of 4 bytes.
* Transformation ensures that the 4 bytes of one column are spread out to four different columns

## Mix Columns Rationale

* Coefficients of a matrix based on a linear code with maximal distance between code words ensures a good mixing among the bytes of each columns
* The mix column transformation combined with the shift row transformation ensures that after a few rounds all output bits depend on all input bits.

## AddRoundKey Transformation

* the 128 bits of State are bitwise XORed with the 128 bits of the round key.
* Operation is viewed as a column wise operation between the 4 bytes of a State column and one word of the round key.
  + Can also be viewed as a byte-level operation
* Rationale:
  + is as simple as possible and affects every bit of State
  + The complexity of the round key expansion plus the complexity of the other stages of AES ensure security

## AES Key Expansion

* takes as input a four-word (16 byte) key and produces a linear array of 44 words (176 bytes).
  + This is sufficient to provide a four-word round key for the initial AddRoundKey stage
* Key is copied into the first four words of the expanded key
  + The remainder of the expanded key is filled in four words at a time
* Each added word w[i] depends on the immediately preceding word, w [i -1], and the word four positions back, w [ i – 4]
  + In three out of four cases a simple XOR is used
  + For a word whose position in the w array is a multiple of 4, a more complex function is used

## Key Expansion Rationale

* The Rijndael developers designed the expansion key algorithm to be resistant to known cryptanalytic attacks
* Inclusion of a round-dependent round constant eliminates the symmetry between the ways in which round keys are generated in different rounds.

Criteria used:

* Knowledge of a part of the cipher key or round key does not enable calculation of many other round-key bits
* An invertible transformation
* Speed on a wide range of processors
* Usage of round constants to eliminate symmetries
* Diffusion of cipher key differences into the round keys
* Enough nonlinearity to prohibit the full determination of round key differences from cipher key differences only
* Simplicity of description

## Equivalent Inverse Cipher

AES decryption cipher is not identical to the encryption cipher.

* the sequence of transformations differs although the form of the key schedules is the same.
* Has the disadvantage that two separate software or firmware are needed for applications that require both encryption and decryption

Two separate changes are needed to bring the decryption structure in line with the encryption structure:

* the first two stages of the decryption round need to be interchanged
* The second two stages of the decryption round need to be interchanged.

## Interchanging: InvShiftRows and InvSubBytes.

* InvShiftRows affects the sequence of bytes in State but does not alter byte contents and does not depend on byte contents to perform its transformation.
* InvSubBytes affects the contents of bytes in State but does not alter byte sequence and does not depend on byte sequence to perform its transformation.
* these two operations commute and can be interchanged.

## Interchanging: AddRoundKey and InvMixColumns

The transformations AddRoundKey and InvMixColumns do not alter the sequence of bytes in State.

If we view the key as a sequence of words, then both AddRoundKey and InvMixColumns operate on State one column at a time.

These two operations are linear with respect to the column input.

## Implementation Aspects

* AES can be implemented very efficiently on a 8-bit processor
* AddRoundKey is a bytewise XOR operation.
* ShiftRows is a simple byte-shifting operation
* SubBytes operates at the byte level and only requires a table of 256 bytes.
* MixColumns requires matrix multiplication in the field GF (28), which means that all operations are carried out on bytes.
* Can efficiently implement on a 32 bit processor
  + Redefine steps to use 32-bit words
  + Can precompute 4 tables of 256 words
  + Then each column in each round can be computed using 4 table lookups + 4 XORs
  + At a cost of 4Kb to store tables.
* Designers believe this very efficient implementation was a key factor in its selection as the AES cipher

# Chapter 9: Public Key Cryptography and RSA

## Misconceptions concerning Public-Key Encryption

* Public key encryption is more secure from cryptanalysis than symmetric encryption.
* Public Key encryption is a general-purpose technique that has made symmetric encryption obsolete (onnodig)
* There is a feeling that key distribution is trivial when using public key encryption, compared to the cumbersome handshaking involved with key distribution centres for symmetric encryption. It is not simpler nor any more efficient than those required for symmetric encryption.

## Principles of Public Key Cryptosystems

The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption:

* key distribution
  + How to have secure communications in general without having to trust a KDC with your key
* digital signatures
  + how to verify that a message comes intact from the claimed sender.

Withfield Diffie And Martin Hellman from Stanford University achieved a breakthrough in 1976 by coming up with a method that addressed both problems and was radically different from all previous approaches to cryptography.

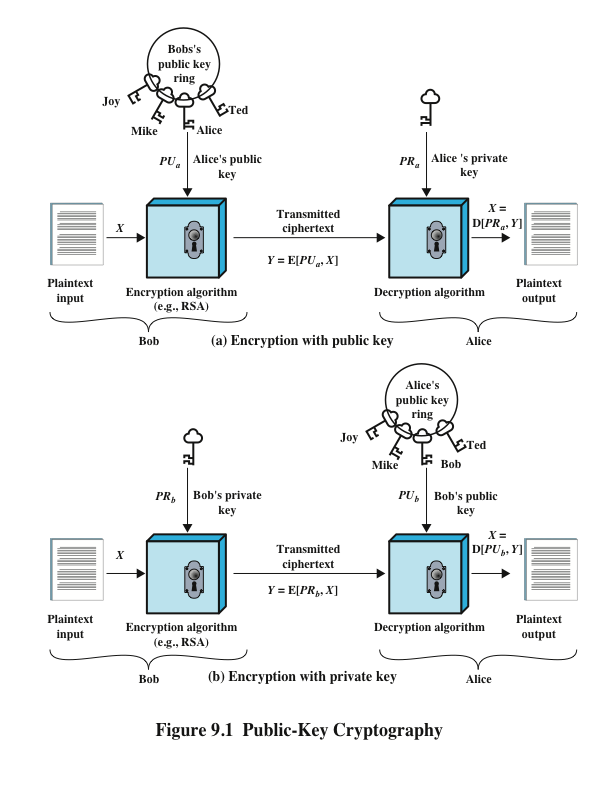
## Public Key Cryptosystems

Six ingredients:

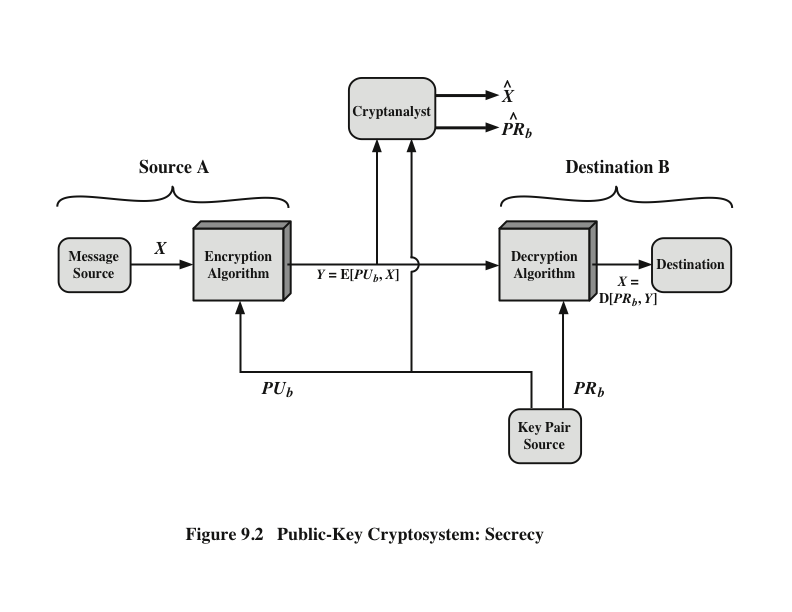
* Plaintext
* Encryption Algorithm
* Public key
* Private key
* Ciphertext
* Decryption Algorithm

Public Key Cryptography STEPS

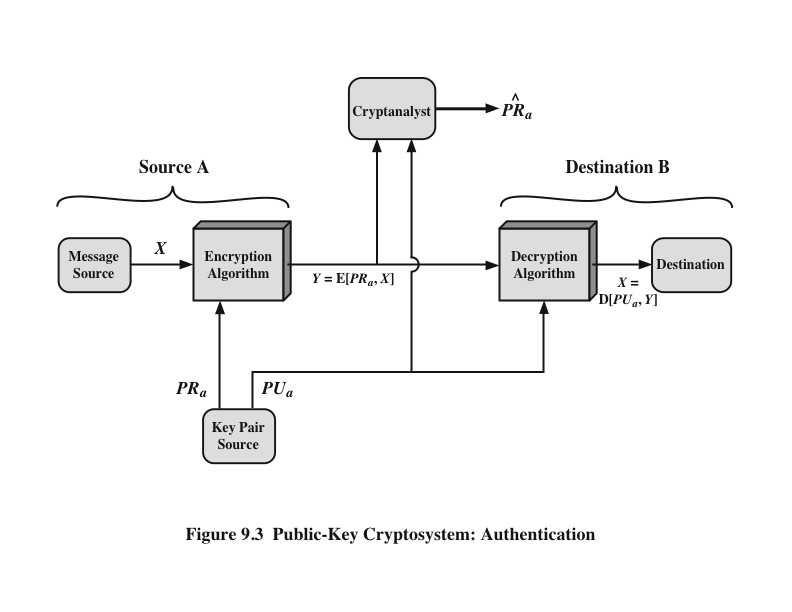
1. Each uses generates a pair of keys to be used for the encryption and decryption of messages.
2. Each users places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
3. If Bob wishes to send a confidential message to Alice, 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 Alice’s private key.



## Public Key Cryptosystem: Secrecy



## Public Key Cryptosystem: Authentication

****

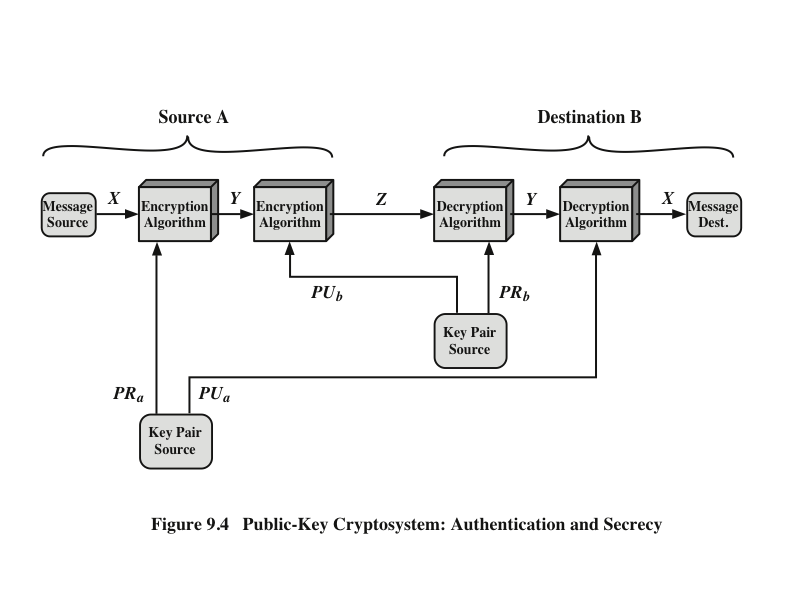
Both keys can be used for encryption, by using the private key as the input for the encryption algorithm and the public key for the encryption of the private key you can provide authentication.

However it does not provide confidentiality. The message is being safe from alteration, but not from eavesdropping.

* any observer can decrypt the message by using the public key of the sender.

## Public Key Cryptosystem: Authentication and Secrecy

It is however possible to provide both the authentication function and confidentiality by a double use of the public key scheme.



In this case:

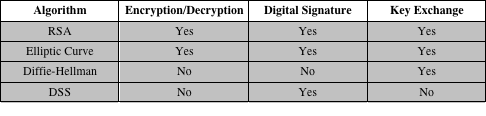
* encrypting a message using the sender’s private key. (provides digital signature).
* Encrypt again using the receiver public key.
* The final ciphertext can be decrypted only by the intended receiver , who alone has the matching private key. 🡺 confidentiality.
* Disadvantage: the public key algorithm, which is complex, must be exercised four times rather than two in each communication.

## Applications for Public-Key Cryptosystems

Three categories:

* Encryption / decryption
  + The sender encrypts a message with the recipient’s public key
* Digital Signature
  + the sender “signs” a message with its private key
* Key Exchange
  + Two sides cooperate to exchange a session key.

Some algorithms are suitable for all three applications, whereas others can be used only for one or two.



## Public Key Requirements

Conditions:

* it is computationally easy for a party B to generate a pair keys
* It is computationally easy for a sender A , knowing the public key and the message to be encrypted, to generate the corresponding ciphertext.
* it is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message.
* It is computationally infeasible for an adversary, knowing the public key, to determine the private key
* It is computationally infeasible for an adversary, knowing the public key and a ciphertext, to recover the original message
* The two keys can be applied in either order.

Need a trap-door one-way function

* a one-way function is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible.
  + Y = f(x) = easy
  + X = f-1 (Y) = infeasible

A trap-door one-way function is a family of invertible functions fk, such that

* Y = fk(X) easy, if k and X are known
* X = fk–1(Y) easy, if k and Y are known
* X = fk–1(Y) infeasible, if Y known but k not known

A practical public-key scheme depends on a suitable trap-door one-way function.

## Public Key Cryptanalysis

A public key encryption scheme is vulnerable to a brute-force attack

* countermeasure: use large keys
* key size must be small enough for practical encryption and decryption
* key sizes that have been proposed result in encryption / decryption speeds that are too slow for general-purpose use
* Public-key encryption is currently confined to key management and signature applications.

Another form of attack is to find some way to compute the private key given the public key

* to date it has not been mathematically proven that this form of attack is infeasible for a particular public-key algorithm

Finally there is a probable-message attack

* this attack can be thwarted by appending some random bits to simple messages.

## Rivest-Shamir-Adleman Scheme (RSA)

Developed in 1977 at MIT by Ron Rivest, Adi Shamir and Len Adleman.

Most widely used general purpose approach to public-key encryption

Is a cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n

* a typical size for n = 1024 bits, or 309 decimal digits.

## RSA Algorithm

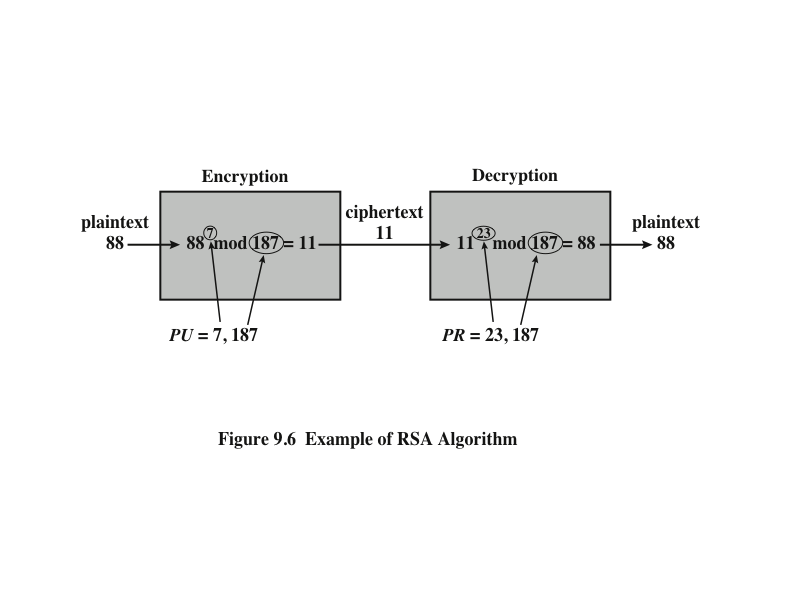
* makes uses of an expression with exponentials
* Plaintext is encrypted in blocks with each block having a binary value less than some number n
* Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C
  + C = Me mod n
  + M = Cd mod n = (Me)d mod n = Med mod n
* Both sender and receiver must know the value of n
* The sender knows the value of e, and only the receiver knows the value of d.
* This is a public –key encryption algorithm with a public key PU = { e, n } and a private key PR = { d, n }

## Algorithm Requirements

For this algorithm to be satisfactory to public key encryption, the following requirements must be met:

* it is possible to find values of e, d, n, such that Med mod n = M for all M < n
* it is relatively easy to calculate Me mod n and Cd mod n for all values M < n
* it is infeasible to determine d given e and n.

## Example of RSA Algorithm



## Exponentiation in Modular Arithmetic

Both encryption and decryption in RSA involve raising an integer to an integer power, mod n

Can make use of a property of modular arithmetic:

[(*a* mod *n) x (b* mod *n)]* mod *n* =(*a x b)* mod *n*

With RSA you are dealing with potentially large exponents of efficiency of exponentiation is a consideration.

## Efficient Operation Using the Public Key

To speed up the operation of the RSA algorithm using the public key, a specific choice of e is usually made.

The most common choice is 65537 (2 16 + 1).

* two other popular choices are e = 3 and e = 17
* Each of these choices has only two 1 bits, so the number of multiplications required to perform exponentiation is minimized.
* With a very small public key, such as e = 3, RSA becomes vulnerable to a simple attack.

## Efficient Operation Using the Private Key

Decryption uses exponentiation to power d

* a small value of d is vulnerable to a brute-force attack and to other forms of cryptanalysis

Can use the Chinese Remainder Theorem (CRT) to speed up computation.

* The quantities d mod (p – 1) and d mod (q -1) can be precalculated.
* End result is that the calculation is approximately four times as fast as evaluating M = Cd mod n directly.

## Key Generation

Before the application of the public-key cryptosystem each participant must generate a pair of keys:

* determine two prime numbers p and q
* select either e or d and calculate the other.

Because the value of n = pq will be known to any potential adversary, primes must be chosen from a sufficiently large set

* the method used for finding large primes must be reasonably efficient.

Procedure for picking a prime number

* pick an odd integer n at random
* pick an integer a < n at random
* perform the probabilistic primality test with a as a parameter. if n fails the test, rejects the value n and go to step 1.
* If n has passed a sufficient number of tests accept n; otherwise, go to step 2

## The Security of RSA

Five possible approaches:

* Chose ciphertext attacks
  + this type of attack exploits properties of the RSA algorithm
* Brute force
  + involves trying all possible private keys
* Mathematical attacks
  + There are several approaches, all equivalent in effort to factoring the product of two primes
* Timing attacks
  + These depend on the running time of the decryption algorithm
* Hardware Fault –based attack
  + This involves inducing hardware faults it the processor that is generating digital signatures.

## Factoring Problem

We can identify three approaches to attacking RSA mathematically:

* factor n into its two prime factors. This enables calculation of ø(*n*) = (*p – 1) x (q* – 1), which in turn enables determination of *d = e-1 (mod* ø(n)).
* Determine ø(n) directly without first determining *p* and *q.* Again this enables determination of *d = e-1 (mod* ø(n))
* Determine *d* directly without first determining ø(n)

## Timing Attacks

Paul Kocher, a cryptographic consultant, demonstrated that a snooper can determine a private key by keeping track of how long a computer takes to decipher messages.

Are applicable not just to RSA but to other public-key cryptography systems.

Are alarming for two reasons:

* it comes from a completely unexpected direction
* it is a ciphertext-only attack

## Countermeasures

* Constant exponentiation time
  + Ensure that all exponentiations take the same amount of time before returning a result, this is a simple fix but does degrade performance
* Random delay
  + Better performance could be achieved by adding a random delay to be exponentiation algorithm to confuse the timing attack
* Blinding
  + Multiply the ciphertext by a random number before performing exponentiation; this process prevents the attacker from knowing what ciphertext bits are being processed inside the computer and therefor prevents the bit-by-bit analysis essential to the timing attack.

## Fault-Based attack

An attack on a processor that is generating RSA digital signatures

* induces faults in the signature computation by reducing the power to the processor
* the faults cause the software to produce invalid signatures which can then be analysed by the attacker to recover the private key.

The attack algorithm involves inducing single-bit errors and observing the results

While worthy of consideration, this attack does not appear to be a serious threat to RSA

* it requires that the attacker have physical access to the target machine and is able to directly control the input power to the processor.

## Chosen Ciphertext attack (CCA)

The adversary chooses a number of ciphertexts and is then given the corresponding plaintexts, decrypted with the target’s private key

* Thus the adversary could select a plaintext, encrypt it with the targets public key, and then be able to get the plaintext back by having it decrypted with the private key
* The adversary exploits properties of RSA and selects blocks of data that, when processed using the targets private key, yield information needed for cryptanalysis.

To counter such attacks, RSA security Inc. recommends modifying the plaintext using a procedure known as optimal asymmetric encryption padding (OAEP)

# Chapter 10: Other Public-Key Cryptosystems

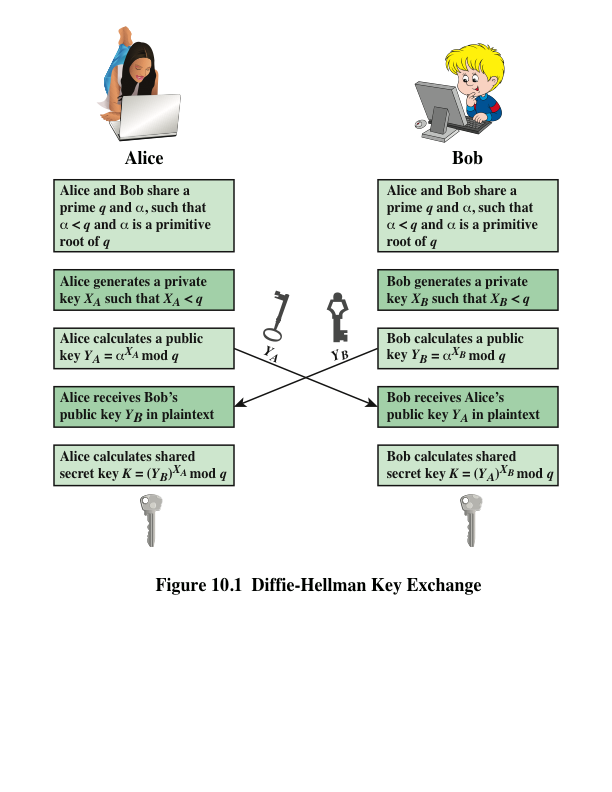
## Diffie-Hellman Key Exchange

A number of commercial products employ this key exchange technique.

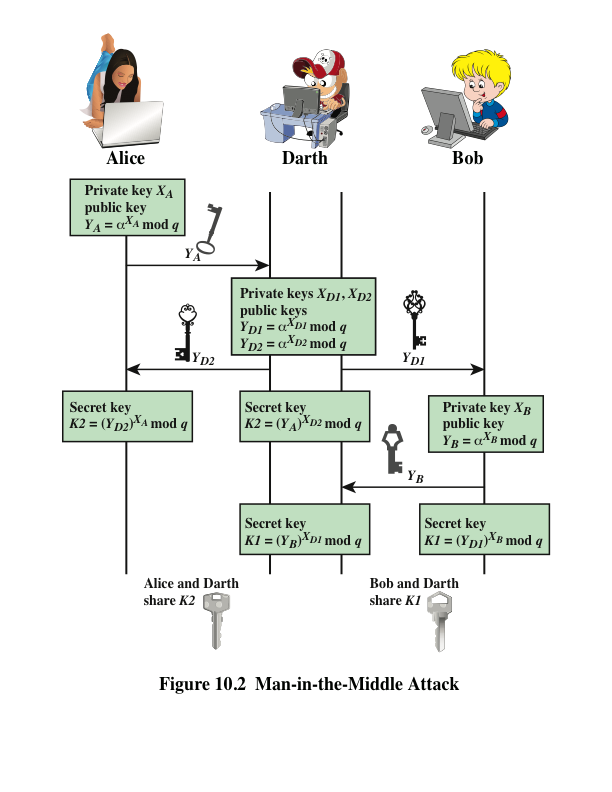
Purpose is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages.

The algorithm itself is limited to the exchange of secret values.

Its effectiveness depends on the difficulty of computing discrete logarithms.



## Key Exchange Protocols



Users could create random private / public Diffie Hellman keys each time they communicate.

Users could create a known private / public Diffie – Hellman key and publish in a directory, then consulted and used to securely communicate with them.

Vulnerable to Man-in-the-middle Attack

Authentication of the keys is needed!

## ElGamal Cryptography

Public-key scheme based on discrete logarithms closely related to the Diffie-Hellman technique.

Used in the digital signature standard (DSS) and the S/MIME e-mail standard.

Global elements are a prime number q and a which is a primitive root of q.

Security is based on the difficulty of computing discrete logarithms.

## Elliptic Curve Arithmetic

Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA.

* the key length for secure RSA use has increased over recent years and this has put a heavier processing load on applications using RSA.

Elliptic curve cryptography (ECC) is showing up in standardization efforts including the IEEE P1363 Standard for Public Key Cryptography

Principal attraction of ECC is that it appears to offer equal security for a far smaller key size.

Confidence level in ECC is not yet as high as that in RSA however.

## Elliptic Curves over Zp

Elliptic curve cryptography uses curves whose variables and coefficients are finite.

Two families of elliptic curves are used in cryptographic applications:

* Binary curves over GF (2m)
  + Variables and coefficients all take on values in GF (2m) and in calculations are performed over GF (2m)
  + Best for hardware applications
* Prime curves over Zp
  + Use a cubic equation in which the variables and coefficients all take on values in the set of integers from 0 through p-1 and in which calculations are performed module p
  + best for software applications

## Elliptic Curves over GF (2m)

Use a cubic equation in which the variables and coefficients all take on values in GF (2m) for some number m.

Calculations are performed using the rules of arithmetic in GF (2m).

The form of cubic equation appropriate for cryptographic applications for elliptic curves is somewhat different for GF (2m) than for Zp

* it is understood that the variables x and y and the coefficients a and b are elements of GF (2m) and that calculations are performed in GF (2m).

## Elliptic Curve Cryptography (ECC)

Addition operation in ECC is the counterpart of modular multiplication in RSA.

Multiple addition is the counterpart of modular exponentiation.

To form a cryptographic system using elliptic curves, we need to find a “hard problem” corresponding to factoring the product of two primes or taking the discrete logarithm.

Q = kP where Q, P belong to a prime curve

* is easy to compute Q given k and P
* But hard to find k given Q and P
  + 🡺 Known as the elliptic Curve logarithm problem.

## ECC Encryption / Decryption

* Several approaches using elliptic curves have been analysed
* Must first encode any message m as a point on the elliptic curve Pm
* Select suitable curve and point G as in Diffie-Hellman
* Each user chooses a private key nA and generates a public key PA = nA \* G
* To encrypt and send message Pm to B, A chooses a random positive integer k and produces the ciphertext Cm consisting of the pair of points:

*Cm = {kG, Pm+kPB}*

* To decrypt the ciphertext, B multiplies the first point in the pair by B’s secret key and subtract eh result from the second point:

*Pm+kPB–nB(kG) = Pm+k(nBG)–nB(kG) = Pm*

## Security of Elliptic Curve Cryptography

* depends on the difficulty of the elliptic curve logarithm problem
* fastest known technique is “Pollard rho method”
* Compared to factoring, can use much smaller key sizes than with RSA
* For equivalent key lengths computations are roughly equivalent
* Hence, for similar security ECC offers significant computational advantages

Pseudorandom Number Generation (PRNG) Based on Asymmetric Cipher

* An asymmetric encryption algorithm produces apparently ransom output and can be used to build a PRNG.
* Much slower than symmetric algorithms so they’re not used to generate open-ended PNRG bit streams.
* useful for creating a pseudorandom function (PRF) for generating a short pseudorandom bit sequence.

PRNG based on Elliptic Curve Cryptography

* Developed by the U.S. National Security Agency (NSA)
* Known as dual elliptic curve PRNG
* Has been some controversy regarding both the security and efficiency of this algorithm compared to other alternatives.
  + the only motivation for its use would be that it is used in a system that already implements ECC but does not implement any other symmetric, asymmetric or hash cryptographic algorithm that could be used to build a PRNG.